Turning Multiple Key-Dependent Attacks into Universal Attacks

No Author Given

No Institute Given

Abstract. Key-dependent attacks are effective only for specific weak-key classes, limiting their practical impact. We present a generic statistical framework that combines multiple key-dependent distinguishers into universal attacks covering the full key space. Using log-likelihood ratio statistics, our framework tests the secret key against multiple weak-key distinguishers, aggregates their evidence to determine whether the key is weak or strong for each distinguisher, and exploits this classification to reduce the effective key entropy for key recovery.

We apply this to Orthros-PRF, a sum-of-permutations (SoP) design where any differential-based distinguisher holds only for a fraction of keys. This yields the first universal 8-round differential-linear (DL) key-recovery attack with median time complexity $2^{119.58}$, whereas prior work reached at most 7 rounds in the weak-key setting.

To discover the required distinguishers, we extend the open-source S-box Analyzer tool with MILP support for deterministic propagation and develop a model integrating distinguisher search with key recovery. This enables automated discovery of multidimensional DL distinguishers covering up to 10 rounds in each Orthros branch, improving prior work by 4 rounds.

Our results demonstrate that statistical aggregation of multiple weak-key distinguishers enables effective universal cryptanalysis. Our framework is generic and is applicable to other primitives with multiple identifiable weak-key classes.

Keywords: Cryptanalysis · Differential-linear attack · Key-dependent attack · Universal attack · Orthros · Sum of permutations

1 Introduction

Some cryptanalytic attacks succeed only for a subset of the key space. We call such attacks *key-dependent attacks*, and we refer to the subsets of keys for which the attack performs (sometimes even better than expected) as *weak-key classes*. Historically, the concept of weak-keys has been known for decades. A well known example is DES, where specific weak-keys make encryption equal to decryption [36,25]. More recently, invariant subspace attacks [32] and nonlinear invariant attacks [42] have been discussed as weak-key attacks. Although these weak-key attacks involve only a small fraction of keys from the full key space, serious vulnerabilities can arise if such weak-keys are used.

Another source of weak-key classes arises from the breakdown of statistical assumptions used in symmetric key cryptanalysis. Most statistical attacks, including differential, boomerang, and differential linear attacks, rely on the Markov cipher assumption and the hypothesis of stochastic equivalence [29]. These assumptions make it possible to estimate the average probability of a distinguisher over all keys, yet they do not always capture the fixed key behavior of a cipher. Knudsen already observed that the probabilities of differential characteristics in `DES` vary greatly across different keys [28]. At CRYPTO 2022, Beyne and Rijmen introduced the quasidifferential framework [10], which makes it possible to compute fixed key differential probabilities without relying on these assumptions. They showed that several proposed differential attacks fail in the fixed key setting or apply only to a restricted subset of keys. In general, when the underlying assumptions of a statistical attack break down, the attack may fail for some subsets of keys while remaining effective, sometimes even more than expected, for others [10,39,37].

In some cases the weak-key class is very small, while in others it can be relatively large. A recent example is the weak-key attack described by Flórez Gutiérrez et al. at ASIACRYPT 2024 [18]. They showed that the differential and DL attacks on Orthros, proposed in [33], are only effective for about $2^{-4.55}$ of the key space. This limitation stems from the sum of permutations (SoP) structure of Orthros [6], which causes the given differential-based attack to hold only for a fraction of keys rather than the entire key space.

Weak-key classes are often not unique. For a given encryption algorithm, there may exist multiple weak-key attacks, each associated with a different weak-key set. In some cases, the structure of the design can even be exploited to derive several weak-key attacks from a single one. For example, certain primitives such as the `Ascon` [16] permutation are invariant with respect to differential and linear trails under specific state rotations.

This observation raises an important question: how can we combine multiple key-dependent attacks to construct a stronger universal attack that succeeds for a larger fraction, or ideally all, of the keys? Even if a given key-dependent attack is suboptimal, others may complement it. By combining several such attacks, we can reduce the effective key entropy and build a universal attack covering a large fraction of the key space. The main contribution of this work is a generic framework that achieves this goal, demonstrated through its application to Orthros.

1.1 Our Contributions

The contributions of this work are as follows:

- We propose a generic framework for combining multiple key-dependent attacks into a universal attack. The framework tests the secret key against multiple weak-key distinguishers using log-likelihood ratio statistics and aggregates the results to reduce effective key entropy across the full key space. An alternative classification method based on the maximum statistic is analyzed in Section D.

– We apply our framework to Orthros-PRF and obtain the first universal 8-round DL key-recovery attack with median time $2^{119.58}$, whereas prior work reached at most 7 rounds in the weak-key setting [24,33]. More precisely, in our 8-round Orthros-PRF attack, the complexity varies across the key space: the median attack time is $2^{119.58}$ (achieved for at least 50% of keys), while the worst-case complexity is $2^{126.92}$ (for approximately 48% of keys that are strong against all employed distinguishers).
– We develop a MILP model for automated discovery of complete DL attacks that integrates distinguisher search with key recovery. We extend the open-source S-box Analyzer to support deterministic propagation of differences and linear masks, and discover multidimensional DL distinguishers covering up to 10 rounds in each Orthros branch, improving the prior best by 4 rounds.
– Table 1 and Table 2 summarize our results compared to prior works, and the experimental observations that motivate and validate our statistical modeling appear in Section E. Moreover, we provide open-source implementations of our attack discovery tools and statistical framework, allowing independent verification and supporting future work on other primitives:

`https://github.com/hadipourh/universalattacks`

1.2 Outline

The rest of this paper is organized as follows. Section 2 provides essential background on statistical cryptanalysis and differential-linear attacks, and formalizes the weak-key problem. Section 3 describes our MILP-based automated search for differential-linear distinguishers and its extension to full key-recovery attacks, and applies it to Orthros to discover multiple multidimensional distinguishers covering up to 10 rounds in each branch. Section 4 develops the statistical framework for distinguishing weak keys from strong keys using log-likelihood ratio statistics for a single distinguisher. Section 5 extends this to combine multiple weak-key distinguishers and presents our universal key-recovery attacks on 6-round and 8-round Orthros. Results are summarized in Table 1 and Table 2. The Orthros specification appears in Section A.7, and empirical correlation signatures that motivate and validate our statistical modeling appear in Section E. Section 6 concludes with remarks and future directions.

2 Background

In this section, we first discuss the problem statement and provide the necessary background for our methodology.

2.1 Problem Statement

Here, we state the problem of weak-key attacks and show that, in some cases, the weak-key issue is inevitable in cryptanalytic attacks. The problem we discuss

Table 1: Summary of distinguishers for Orthros. D: Differential, DL: Differential-Linear, T-D: Truncated-Differential. CP: Chosen Plaintext and $*$ marks weak-key distinguishers.

Branch1				**Branch2**			
Distinguisher	Rounds	Data (CP)	Ref.	Distinguisher	Rounds	Data (CP)	Ref.
Integral	7	2^{127}	[4]	Integral	7	2^{127}	[4]
D	9	2^{113}	[41]	D	9	2^{117}	[41]
DL	6	2^{46}	[33]	DL	6	2^{46}	[33]
DL	6	$\mathbf{2^{12}}$	Section 3.1	DL	6	$\mathbf{2^{12}}$	Section 3.1
DL	**10**	$2^{81.50}$	Section 3.1	DL	**10**	$2^{74.54}$	Section 3.1

PRF			
Distinguisher	Rounds	Data (CP)	Ref.
Integral	7	2^{127}	[4]
D$*$	7	$2^{116.8}$	[41]
DL$*$	7	$\mathbf{2^{84.88}}$	Section 3.2

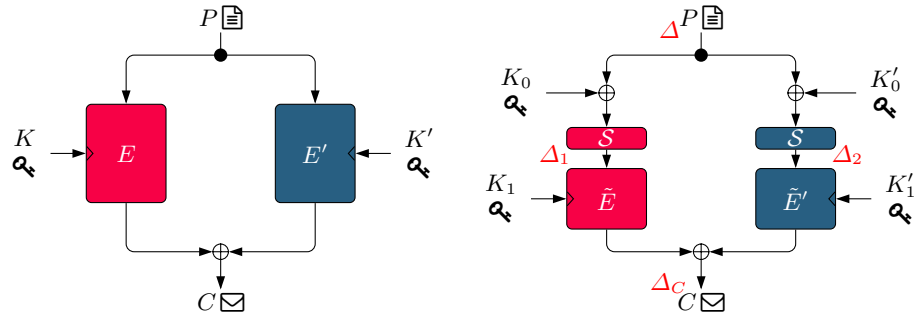
Table 2: Summary of key-recovery attacks on Orthros-PRF.

Attack	Setting	Rounds	Time	Data (CP)	Memory	Coverage	Reference
D	Weak-Key	6	2^{106}	2^{95}	2^{29}	$2^{-4.55}$	[33]
DL	**Universal**	6	$2^{115.07}$	$\mathbf{2^{40}}$	2^{40}	**1**	Section 5.1
DL	Weak-Key	7	2^{120}	2^{115}	2^{29}	$2^{-4.55}$	[33]
T-D	Weak-Key	7	$2^{117.03}$	$2^{75.06}$	$2^{76.07}$	2^{-5}	[24]
DL	**Universal**	**8**	$2^{119.58}$	2^{112}	2^{112}	**1**	Section 5.2

in this section was initially introduced at ASIACRYPT 2024 [18]. Let $E, E' : \mathbb{F}_2^k \times \mathbb{F}_2^n \to \mathbb{F}_2^n$ be two SPN block ciphers with the same block size n and key size k. Figure 1a depicts the overall structure of a pseudorandom function (PRF) following the sum-of-permutations (SoP) design [4,15,19,14]. This PRF is defined as $C = E_K(P) \oplus E'_{K'}(P)$, where P is the plaintext, C is the ciphertext, and K, K' are branch keys derived from the master key by the key schedule. The addition is performed over $\mathbb{F}_2^n$. Note that K and K' are not necessarily equal, although they are related.

Let $\tilde{E}$ and $\tilde{E}'$ be the reduced-round versions of E and E', respectively. As shown in Figure 1b, assume that we want to apply a differential attack based on a reduced-round differential distinguisher for $\tilde{E}_{K_1}(X) \oplus \tilde{E}'_{K'_1}(X)$. As illustrated in Figure 1b, assume that the distinguisher starts after the first round, and let (Δ_1, Δ_2) be the input and output differences of the reduced-round distinguisher.

Moreover, assume that K_0 and K'_0 are the subkeys used before the first S-box layer. That is, K_0 and K'_0 are not necessarily equal, but they must satisfy the relation imposed by the key schedule, which we denote by $\mathcal{KS}(K_0, K'_0)$. For any plaintext P and any input difference Δ, the following conditions must hold for



(a) Black-box view (b) Distinguisher and key-recovery

Fig. 1: Weak-key attack problem on sum of permutations.

the distinguisher to be valid:

$$\begin{cases} S(P \oplus K_0) \oplus S(P \oplus K_0 \oplus \Delta) = \Delta_1, \\ S(P \oplus K'_0) \oplus S(P \oplus K'_0 \oplus \Delta) = \Delta_2, \end{cases} \tag{1}$$

where $P, K_0, K'_0, \Delta \in \mathbb{F}_2^n$.

It is easy to see that if $K'_0 = K_0$ and $\Delta_1 \neq \Delta_2$, then the two conditions in Equation 1 contradict each other, and the distinguisher does not hold.

Let $\mathcal{X}_{\texttt{DDT}}(\Delta_{\text{I}}, \Delta_{\text{O}})$ denote the set of input values that satisfy the differential transition $\Delta_{\text{I}} \to \Delta_{\text{O}}$ through the S-box, i.e.,

$$\mathcal{X}_{\texttt{DDT}}(\Delta_{\text{I}}, \Delta_{\text{O}}) = \{X \in \mathbb{F}_2^n \mid S(X) \oplus S(X \oplus \Delta_{\text{I}}) = \Delta_{\text{O}}\}.$$

If $K_0 \neq K'_0$, then the two conditions in Equation 1 and the relations induced by the key schedule imply[1]:

$$\begin{cases} K_0 \in \mathcal{X}_{\texttt{DDT}}(\Delta, \Delta_1) \oplus P, \\ K'_0 \in \mathcal{X}_{\texttt{DDT}}(\Delta, \Delta_2) \oplus P, \\ \mathcal{KS}(K_0, K'_0) = \text{True}. \end{cases} \tag{2}$$

Although P and Δ are fully under the attacker's control, for any choice of P and Δ, it may happen that either $\mathcal{X}_{\texttt{DDT}}(\Delta, \Delta_1)$ or $\mathcal{X}_{\texttt{DDT}}(\Delta, \Delta_2)$ is empty, or only a subset of keys satisfies Equation 2. We refer to these keys as *weak-keys*.

Definition 1 (Weak-keys and Strong-keys). *For a given distinguisher, the set of keys for which the distinguisher is valid is called the set of weak-keys. Conversely, the set of keys for which the distinguisher does not hold is called the set of strong-keys. That is, under a strong-key, the cipher behaves indistinguishably from an ideal cipher with respect to the given distinguisher.*

[1] For $S \subseteq \mathbb{F}_2^n$, and $X \in \mathbb{F}_2^n$, we define $S \oplus X = \{s \oplus X \mid s \in S\}$.

The set of weak-keys can be described either by a set of constraints or by explicitly listing its elements. For a given weak-key, there are only a certain set of input pairs for which the distinguisher holds. Each weak-key is thus associated with a set of *good plaintexts* or *good pairs*, which are specific input values that produce an output behavior consistent with the distinguisher's conditions. For example, if (K_0, K_0') is a weak-key pair in Figure 1b, its corresponding good pairs $(P, P \oplus \Delta)$ should satisfy:

$$P \in (\mathcal{X}_{\texttt{DDT}}(\Delta, \Delta_1) \oplus K_0) \cap (\mathcal{X}_{\texttt{DDT}}(\Delta, \Delta_2) \oplus K_0') \,. \tag{3}$$

Since different weak-keys may support different sets of good pairs, the weak-key space naturally partitions into distinct classes.

Definition 2 (Weak-Key Classes). *For a distinguisher, a weak-key class is a set of weak-keys that share the same set of good input assignments. A single distinguisher typically yields multiple weak-key classes $\mathcal{W}_1, \mathcal{W}_2, \ldots, \mathcal{W}_\ell$, where each class $\mathcal{W}_i$ is associated with a distinct set of good input assignments $\mathcal{X}_i$.*

Example 1 (Weak-Key Classes). Consider Figure 1b with a concrete instantiation where $S = S_0 \| S_1$ and both S_0 and S_1 are instances of the Orthros S-box. Let $\Delta_1 = \texttt{0x44}$ and $\Delta_2 = \texttt{0x22}$ be the output differences of $S = S_0 \| S_1$ in Figure 1b. Suppose the key bits are $K_0 = (\boldsymbol{k_0}, k_1, k_2, k_3, \boldsymbol{k_4}, k_5, \boldsymbol{k_6}, k_7)$ for the left branch and $K_0' = (\boldsymbol{k_4}, k_8, \boldsymbol{k_0}, k_9, k_{10}, \boldsymbol{k_6}, k_{11}, k_{12})$ for the right branch. Since 3 key bits are shared between branches, the total involved key space takes 2^{13} possible values. In this configuration, we identify 1568 weak-keys (38.3% of the key space), which partition into 504 distinct weak-key classes, each indexed by its sorted list of (unordered) good pairs $\mathcal{X}_i$. Figure 8 illustrates the distribution of weak-key class sizes. For instance, one particular weak-key class $\mathcal{W}_i$ contains a single weak-key $(K_0, K_0') = (\texttt{0xf9}, \texttt{0xea})$ with the following 4 good unordered pairs:

$$\mathcal{X}_i = \{(\texttt{0x30}, \texttt{0x52}), (\texttt{0x32}, \texttt{0x50}), (\texttt{0x40}, \texttt{0x62}), (\texttt{0x42}, \texttt{0x60})\}.$$

This example demonstrates that different weak-key classes can vary significantly in both the number of weak-keys and good pairs they contain.

The basic example discussed above demonstrates that the problem of weak-key attacks is inherent in differential-based attacks on SoP designs. However, this issue is not limited to SoP structures or to differential attacks alone; it can also appear in other cipher designs and attack types [38,7,22,26,27,35,34], including weak-key attacks such as invariant and nonlinear invariant subspace attacks [31,32,42,8,43,9].

2.2 Correlation and Capacity of Boolean Functions

We introduce the key statistical measures for analyzing Boolean functions and establish their mathematical relationships, which underpin our classification framework.

Definition 3 (Correlation). *The correlation of a Boolean function* $f : \mathbb{F}_2^n \to \mathbb{F}_2$ *is defined as:* $\mathrm{Cr}(f) \coloneqq 2^{-n} \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)}$. *The bias is* $\mathcal{E}(f) = \frac{\mathrm{Cr}(f)}{2}$. *The empirical correlation over a sample* $\mathcal{S}$ *is* $\mathrm{Cr}_{\mathcal{S}}(f) = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} (-1)^{f(x)}$.

We can see that $\mathrm{Cr}(f) = \Pr(f(X) = 0) - \Pr(f(X) = 1)$, where X is uniformly distributed over $\mathbb{F}_2^n$, denoted $X \sim \mathrm{Uniform}(\mathbb{F}_2^n)$.

Definition 4 (Kullback-Leibler Divergence or Relative Entropy). *Assume that* X *and* Y *are two random variables with probability mass functions* $p(x)$ *and* $q(x)$*, respectively. The Kullback-Leibler (KL) divergence between* X *and* Y *is defined as:*

$$D_{KL}(p \parallel q) \coloneqq \sum_x p(x) \log \frac{p(x)}{q(x)},$$

with the convention that $0 \log(0/q) = 0$ *for* $q \geq 0$*, and* $p \log(p/0) = \infty$ *for* $p > 0$.

Definition 5 (Capacity Between Two Random Variables). *Let* X *and* Y *be two discrete random variables with a sample space of size* 2^n*, and probability mass functions* $p(x)$ *and* $q(x)$*, respectively.*

$$\mathrm{Cp}(X, Y) = \sum_x \frac{(p(x) - q(x))^2}{q(x)} = \left(\sum_x \frac{(p(x))^2}{q(x)} \right) - 1,$$

It is also denoted by $\mathrm{Cp}(p, q)$*. If* $Y \sim$ *Uniform*$(range(X))$*, we denote* $\mathrm{Cp}(p, q)$ *by* $\mathrm{Cp}(p)$ *and refer to it as the capacity of* X*:*

$$\mathrm{Cp}(X) = 2^n \sum_x (p(x) - 2^{-n})^2.$$

The $\mathrm{Cp}(X, Y)$ is also known as the χ^2 distance between p and q, and it is minimized when $p(x) = q(x)$ for all x. The capacity $\mathrm{Cp}(X)$ measures how far the distribution of X deviates from uniform. Each term $(p(x) - 2^{-n})^2$ captures the squared deviation of $p(x)$ from the uniform value 2^{-n}, so the summation quantifies the total deviation across all $x \in \mathbb{F}_2^n$. The scaling factor 2^n normalizes the measure to the range $[0, 2^n - 1]$, facilitating comparison across spaces of different sizes. If $X \sim \mathrm{Uniform}(\mathbb{F}_2^n)$, then $p(x) = 2^{-n}$ for all x, yielding $\mathrm{Cp}(X) = 0$. At the other extreme, if the distribution is completely concentrated on a single value, i.e., $p(x) = 1$ for some $x \in \mathbb{F}_2^n$, then $\mathrm{Cp}(X) = 2^n - 1$. Thus, values of $\mathrm{Cp}(X)$ near zero indicate a nearly uniform distribution, while higher values indicate concentration or bias.

Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ be a vectorial Boolean function. By the probability distribution of F, we mean the distribution of $F(X)$ where $X \sim \mathrm{Uniform}(\mathbb{F}_2^n)$. Lemma 1 establishes the Fourier relationship between this output distribution and the correlations of all linear combinations $a \cdot F$.

Lemma 1 (Relation Between Probability Distribution and Correlation [23]). *Let* $X \sim$ *Uniform*$(\mathbb{F}_2^n)$ *and let* $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ *be a vectorial Boolean*

function. Then, we have:

$$p_y = \Pr\left(F(X) = y\right) = 2^{-m} \sum_{a \in \mathbb{F}_2^m} (-1)^{a \cdot y} \mathrm{Cr}(a \cdot F), \text{ for all } y \in \mathbb{F}_2^m, \tag{4}$$

$$\boldsymbol{c}_a = \mathrm{Cr}(a \cdot F) = \sum_{y \in \mathbb{F}_2^m} (-1)^{a \cdot y} \Pr(F(X) = y), \text{ for all } a \in \mathbb{F}_2^m, \tag{5}$$

where $a \cdot y = \sum_{i=1}^{m} a_i y_i$ denotes the inner product over $\mathbb{F}_2$.

For the proof, refer to Section A.3

Theorem 1 (Relation Between Capacity and Correlation). *Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ and let $X \sim$ Uniform$(\mathbb{F}_2^n)$. Then, the capacity of $F(X)$ is given by*

$$\mathrm{Cp}(F(X)) = 2^m \sum_{y \in \mathbb{F}_2^m} (\Pr(F(X) = y) - 2^{-m})^2 = \sum_{a \in \mathbb{F}_2^m \setminus \{0\}} \mathrm{Cr}^2(a \cdot F).$$

For the proof, refer to Section A.5.

2.3 Information-Theoretic Bound on Data Complexity

A fundamental problem in cryptanalysis is to distinguish between two alternative hypotheses, for example, whether a given ciphertext has been encrypted using a real cipher or an ideal one. Such problems fall under the general framework of hypothesis testing. In the simplest case, the task is to distinguish between two independent and identically distributed (i.i.d.) distributions.

Definition 6 (Binary Hypothesis Testing). *Let $X_1, \ldots, X_N$ be a sequence of i.i.d. random variables over a domain Ω. The binary hypothesis testing problem is to decide between:*

- *H_0 (null hypothesis): $X_1, \ldots, X_N \sim q$,*
- *H_1 (alternative hypothesis): $X_1, \ldots, X_N \sim p$,*

where p and q are two distinct probability distributions over Ω.

Let $g(x_1, \ldots, x_N)$ denote the decision rule, where $g(x_1, \ldots, x_N) = 1$ indicates acceptance of H_0, and $g(x_1, \ldots, x_N) = 0$ indicates acceptance of H_1. Since g is a binary function, the test can equivalently be described by the acceptance region $\mathcal{A} \subseteq \Omega^N$, defined as $\mathcal{A} = \{(x_1, \ldots, x_N) \in \Omega^N : g(x_1, \ldots, x_N) = 1\}$. The complement $\mathcal{A}^c$ is referred to as the rejection region.

In hypothesis testing, two types of errors may occur: a *type I error* (rejecting H_0 when it is true), with rate $\alpha = \Pr(\text{decide } H_1 \mid H_0) = \Pr(X \in \mathcal{A}^c \mid H_0)$; and a *type II error* (failing to reject H_0 when H_1 is true), with rate $\beta = \Pr(\text{decide } H_0 \mid H_1) = \Pr(X \in \mathcal{A} \mid H_1)$. The parameter α is the *significance level*, and $1 - \beta$ is the *power* of the test. With equal priors ($\pi_0 = \pi_1 = 1/2$) and 0–1 loss (each mistake has unit cost), the average error probability is $P_e = \frac{1}{2}(\alpha + \beta)$; more generally, $P_e = \pi_0\, \alpha + \pi_1\, \beta$.

Let us denote the action of the distinguisher by $H_b \leftarrow \mathcal{D}$, where $b \in \{0,1\}$ is the decision made by $\mathcal{D}$. The advantage of the distinguisher is defined as

$$\mathrm{Adv}(\mathcal{D}) = \left|\Pr(H_1 \leftarrow \mathcal{D} \mid H_1) - \Pr(H_1 \leftarrow \mathcal{D} \mid H_0)\right|.$$

Assuming w.l.o.g. that $P_e \leq 1/2$, it follows that $\mathrm{Adv}(\mathcal{D}) = 1 - 2P_e$.

The objective in hypothesis testing is to minimize the error probability P_e. For fixed data and a fixed test statistic, the error rates α and β exhibit a fundamental tradeoff: decreasing the Type I error rate α (by raising the threshold) increases the Type II error rate β, and vice versa. One typically constrains α below a target significance level and minimizes β subject to that constraint. Reducing both α and β simultaneously requires either increasing the sample size or employing a more informative test statistic. The Neyman-Pearson lemma establishes that the likelihood ratio test is optimal in this sense: for any fixed significance level α, it achieves the minimum possible Type II error β.

Lemma 2 (Neyman-Pearson Lemma [13]). *Let $X_1, \ldots, X_N$ be i.i.d. random variables following either distribution p (under H_1) or distribution q (under H_0). Define the joint distributions $p(x_1, \ldots, x_N) = \prod_{i=1}^{N} p(x_i)$ and $q(x_1, \ldots, x_N) = \prod_{i=1}^{N} q(x_i)$, and the likelihood ratio as $\mathrm{LR}(x_1, \ldots, x_N) = \frac{p(x_1,\ldots,x_N)}{q(x_1,\ldots,x_N)}$. For any threshold $\tau \geq 0$, consider the likelihood ratio test with acceptance region*

$$\mathcal{A}_\tau = \{(x_1, \ldots, x_N) \in \Omega^N : \mathrm{LR}(x_1, \ldots, x_N) < \tau\},$$

which accepts H_0 if $\mathrm{LR}(x_1, \ldots, x_N) < \tau$ and rejects otherwise. Let $\alpha_\tau = \Pr(\mathcal{A}_\tau^c \mid H_0)$ and $\beta_\tau = \Pr(\mathcal{A}_\tau \mid H_1)$ denote its Type I and Type II error rates. Then, for any other test with acceptance region $\mathcal{B}$ satisfying $\Pr(\mathcal{B}^c \mid H_0) \leq \alpha_\tau$, we have $\Pr(\mathcal{B} \mid H_1) \geq \beta_\tau$.

The Neyman-Pearson lemma establishes that the likelihood ratio test is optimal: among all tests with Type I error at most α_τ, it achieves the minimum Type II error β_τ. Importantly, this optimality holds regardless of computational complexity, making the likelihood ratio test a fundamental benchmark for distinguishers.

Since the random variables $X_1, \ldots, X_N$ are i.i.d., the likelihood ratio factorizes as

$$\mathrm{LR}(x_1, \ldots, x_N) = \prod_{i=1}^{N} \frac{p(x_i)}{q(x_i)} = \exp\left(\sum_{i=1}^{N} \log \frac{p(x_i)}{q(x_i)}\right).$$

Since the logarithm is monotonic and preserves the ordering, the test can be performed using the *log-likelihood ratio* (LLR):

$$\mathrm{LLR}(x_1, \ldots, x_N) = \sum_{i=1}^{N} \log \frac{p(x_i)}{q(x_i)}, \text{ where } \log \frac{0}{q} = -\infty \text{ and } \log \frac{p}{0} = +\infty \text{ for } p > 0.$$

Example 2. For Gaussian distributions $p = \mathcal{N}(\mu_0, \sigma_0^2)$ and $q = \mathcal{N}(\mu_1, \sigma_1^2)$, the log-likelihood ratio on N i.i.d. samples is

$$\mathrm{LLR}(x_1, \ldots, x_N) = N \log\left(\frac{\sigma_1}{\sigma_0}\right) + \sum_{i=1}^{N} \left(\frac{(x_i - \mu_1)^2}{2\sigma_1^2} - \frac{(x_i - \mu_0)^2}{2\sigma_0^2}\right).$$

In practice, the log-likelihood ratio can be computed incrementally by maintaining a running sum $S \leftarrow S + \log \frac{p(x_i)}{q(x_i)}$ as samples arrive, avoiding the need to store all N observations. When the sample space Ω is finite, it is more efficient to maintain counts $c(x)$ for each $x \in \Omega$ and work with the empirical distribution

$$\widehat{p}_N(x) = \frac{c(x)}{N} = \frac{1}{N} \sum_{i=1}^{N} \delta_x(x_i),$$

where $\delta_x(x_i) = 1$ if $x_i = x$ and 0 otherwise. The LLR can then be expressed in histogram form (see Lemma 5 in Section A.4):

$$\mathrm{LLR}_N(\widehat{p}_N, p, q) = N \sum_{x \in \Omega} \widehat{p}_N(x) \log \frac{p(x)}{q(x)}. \tag{6}$$

This representation requires only $\mathcal{O}(|\Omega|)$ memory and is particularly advantageous when N is large or when the same counts are reused across multiple hypothesis tests. In our application, where $|\Omega| = 2^m$ is small and counts are reused for evaluating multiple weak-key classes, this histogram form is the natural choice.

Theorem 2 (Asymptotic Distribution of the LLR [3,13]). *Let $X_1, \ldots, X_N$ be i.i.d. random variables over Ω following either distribution p (under H_1) or distribution q (under H_0). Assume either that Ω is finite with $p(x), q(x) > 0$ for all x, or more generally that the variances of $\log \frac{p(X)}{q(X)}$ under p and of $\log \frac{q(X)}{p(X)}$ under q are finite. Let $\widehat{p}_N(x)$ denote the empirical distribution. Then, as $N \to \infty$, the log-likelihood ratio*

$$\mathrm{LLR}_N(\widehat{p}_N, p, q) = N \sum_{x \in \Omega} \widehat{p}_N(x) \log \frac{p(x)}{q(x)}$$

is asymptotically normally distributed:

- *Under H_1 (data from p):* $\mathrm{LLR}_N \sim \mathcal{N}(N\mu_0, N\sigma_0^2)$, *where*

$$\mu_0 = D_{KL}(p \parallel q), \qquad \sigma_0^2 = \sum_x p(x) \left(\log \frac{p(x)}{q(x)} \right)^2 - \mu_0^2.$$

- *Under H_0 (data from q):* $\mathrm{LLR}_N \sim \mathcal{N}(N\mu_1, N\sigma_1^2)$, *where*

$$\mu_1 = -D_{KL}(q \parallel p), \qquad \sigma_1^2 = \sum_x q(x) \left(\log \frac{q(x)}{p(x)} \right)^2 - \mu_1^2.$$

Moreover, when the distributions are close (i.e., $|p(x) - q(x)| \ll q(x)$ for all $x \in \Omega$), we have $\mu_0 - \mu_1 \approx \mathrm{Cp}(p, q)$ and $\sigma_0^2 \approx \sigma_1^2 \approx \mathrm{Cp}(p, q)$.

Theorem 3 (Information-Theoretic Data Complexity [3,23]). *Consider the binary hypothesis testing problem with $H_0 : X \sim q$ and $H_1 : X \sim p$, where $X_1, \ldots, X_N$ are i.i.d. samples from Ω. Let the distinguisher $\mathcal{D}$ use the likelihood ratio test: accept H_1 if $\mathrm{LLR}_N(\widehat{p}_N, p, q) \geq \tau$, otherwise accept H_0.*

Under the asymptotic normality of LLR_N (Theorem 2), the number of samples needed to achieve type I error α (false positive rate) and type II error β (false negative rate) is approximately

$$N \approx \frac{d}{2D_{KL}(p \parallel q)}, \tag{7}$$

where $d = (z_\alpha - z_\beta)^2$, with $z_\alpha = \Phi^{-1}(\alpha)$, $z_\beta = \Phi^{-1}(1-\beta)$, and Φ denoting the cumulative distribution function (CDF) of the standard normal distribution. Under this approximation with equal priors and 0–1 loss, the average error probability is

$$P_e = \tfrac{1}{2}(\alpha + \beta) \approx \Phi\left(-\tfrac{\sqrt{d}}{2}\right).$$

Moreover, when the distributions are close (i.e., $|p(x) - q(x)| \ll q(x)$ for all $x \in \Omega$), the KL divergence satisfies $D_{KL}(p \parallel q) \approx \frac{1}{2}\mathrm{Cp}(p, q)$ (Lemma 7), and the sample complexity simplifies to

$$N \approx \frac{d}{\mathrm{Cp}(p, q)}. \tag{8}$$

As shown in Theorem 3, the data complexity of the distinguisher is inversely proportional to the KL divergence between the two distributions. This aligns with the intuition that distinguishing between two very similar distributions requires a larger number of samples.

Suppose we wish to distinguish between the distribution of some data derived from a cipher, such as the output distribution of a linear approximation (with probability mass function (PMF) p), and a uniform distribution (with PMF q). Using Theorem 1 together with Theorem 3, we can estimate the number of samples required for such a distinguishing task.

Corollary 1. *Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ be a vectorial Boolean function, and let $X \sim$ Uniform$(\mathbb{F}_2^n)$. Then, using Equation 8 with the capacity formula from Theorem 1, the number of samples required to distinguish the output distribution $F(X)$ from the uniform distribution over $\mathbb{F}_2^m$, with type I error α and type II error β, is estimated as*

$$N \approx \frac{(z_\alpha - z_\beta)^2}{\sum_{a \in \mathbb{F}_2^m,\ a \neq 0} \mathrm{Cr}^2(a \cdot F)}.$$

2.4 Multidimensional Differential-Linear Distinguisher.

Here, we review the differential-linear (DL) distinguisher originally proposed by Langford and Hellman at CRYPTO 1994 [30]. The core idea is to combine a short but strong differential with a short but strong linear distinguisher to construct a

Algorithm 1: DL distinguisher

Input: $E_k, (\delta, \omega), N, \tau$
Output: Distinguishes Between E_k and an Ideal Cipher

Initialize counts $c[0] \leftarrow 0, c[1] \leftarrow 0$;
forall $i = 0, \ldots, N-1$ **do**
  $P_1 \xleftarrow{\$} \mathbb{F}_2^n, \quad C_1 \leftarrow E_k(P_1)$;
  $P_2 \leftarrow P_1 \oplus \delta, \quad C_2 \leftarrow E_k(P_2)$;
  $b \leftarrow \omega \cdot (C_1 \oplus C_2)$;
  $c[b] \leftarrow c[b] + 1$;
$\widehat{p}_N(b) \leftarrow c[b]/N$ for $b \in \{0, 1\}$;
$LLR \leftarrow \mathrm{LLR}_N(\widehat{p}_N, p_{\mathrm{real}}, p_{\mathrm{ideal}})$;
if $LLR \geq \tau$ **then**
  return 0;
else
  return 1;

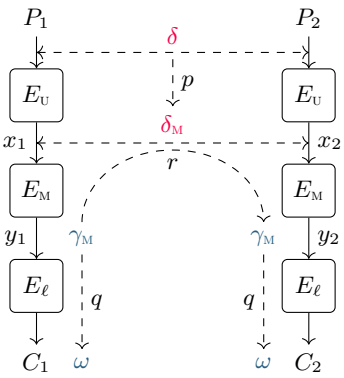


Fig. 2: Building blocks of a DL distinguisher in the sandwich framework.

longer distinguisher that often covers more rounds than achievable using purely differential or purely linear distinguishers.

Algorithm 1 briefly describes the DL distinguisher. The distinguisher queries pairs of plaintexts (P_1, P_2) with difference δ and computes the bit $b = \omega \cdot C_1 \oplus \omega \cdot C_2$ for the corresponding ciphertexts. After collecting N samples, it uses the histogram form of the log-likelihood ratio to distinguish between the real cipher and an ideal cipher based on the empirical distribution $\widehat{p}_N(b)$.

In what follows, we analyze the expected distribution of b under both the real and ideal cipher assumptions and derive the data complexity of the DL distinguisher.

Definition 7 (Autocorrelation). *Let $E_k : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a block cipher with n-bit block size. Given an input difference δ, let $\mathbf{c}_\omega(\delta)$ be the correlation of the DL distinguisher (also referred to as the autocorrelation of E_k) with respect to an output mask ω, defined as follows:*

$$\begin{aligned}\mathbf{c}_\omega(\delta) \coloneqq \mathrm{Cr}(\delta, \omega) &= 2^{-n} \sum_{x \in \mathbb{F}_2^n} (-1)^{\omega \cdot (E_k(x) \oplus E_k(x \oplus \delta))} \\ &= 2 \cdot \Pr\left(\omega \cdot (E_k(X) \oplus E_k(X \oplus \delta)) = 0\right) - 1,\end{aligned}$$

where $X \sim$ Uniform$(\mathbb{F}_2^n)$.

Analytical Estimation of Autocorrelation To analytically estimate the correlation of the DL distinguisher, we often use the sandwich framework [17] together with the generalized differential-linear connectivity table (DLCT) framework [5,20]. In the sandwich framework the block cipher E is divided into three parts: E_{U}, E_{M}, and E_ℓ, as illustrated in Figure 2. While E_{U} and E_ℓ are treated

as standard differential and linear distinguishers, respectively, E_{M} is regarded as a compact combined distinguisher connecting the two segments (see Figure 2).

According to the `DLCT` framework [5,11,20], the correlation of the DL distinguisher is estimated by:

$$\mathrm{Cr}\left(\omega \cdot c_1 \oplus \omega \cdot c_2\right) \approx \sum_{\delta_{\mathrm{M}}, \gamma_{\mathrm{M}}} \Pr\left(\delta, \delta_{\mathrm{M}}\right) \cdot \mathrm{Cr}(\delta_{\mathrm{M}}, \gamma_{\mathrm{M}}) \cdot \mathrm{Cr}^2(\gamma_{\mathrm{M}}, \omega), \tag{9}$$

where $\Pr(\delta, \delta_{\mathrm{M}}) = \Pr(E_{\mathrm{U}}(P) \oplus E_{\mathrm{U}}(P \oplus \delta) = \delta_{\mathrm{M}})$, with P chosen uniformly at random from $\mathbb{F}_2^n$, $\mathrm{Cr}(\delta_{\mathrm{M}}, \gamma_{\mathrm{M}}) = \mathrm{Cr}(\gamma_{\mathrm{M}} \cdot (E_{\mathrm{M}}(X) \oplus E_{\mathrm{M}}(X \oplus \delta_{\mathrm{M}})))$, with X chosen uniformly at random from $\mathbb{F}_2^n$, and $\mathrm{Cr}(\gamma_{\mathrm{M}}, \omega) = \mathrm{Cr}(\gamma_{\mathrm{M}} \cdot E_\ell(Y) \oplus \omega \cdot E_\ell(Y))$, with Y chosen uniformly at random from $\mathbb{F}_2^n$.

Given the input difference δ and the output mask ω for a block cipher E, assume that for a fixed δ_{M} and a fixed γ_{M}, we have $p = \Pr\left(\delta, \delta_{\mathrm{M}}\right)$, $r = \mathrm{Cr}\left(\delta_{\mathrm{M}}, \gamma_{\mathrm{M}}\right)$, and $q = \mathrm{Cr}\left(\gamma_{\mathrm{M}}, \omega\right)$. If E is appropriately decomposed into three segments: E_{U}, E_{M}, and E_ℓ (particularly when E_{M} is long enough), and if δ_{M} and γ_{M} are well-chosen, then prq^2 provides a good estimate of the actual correlation of the DL distinguisher with input difference δ and output mask ω.

Distribution of the Autocorrelation and Data Complexity Let B be the random variable representing the bit $\omega \cdot (C_1 \oplus C_2)$ in Algorithm 1 for a random pair $((P_1, C_1), (P_2, C_2))$ with $P_1 \oplus P_2 = \delta$. From Definition 7, we have $B \sim \mathrm{Bernoulli}(p)$, where $p = \Pr(B = 0) = (1 + \boldsymbol{c}_\omega(\delta))/2$, with $\mathbb{E}[B] = p$ and $\mathrm{Var}(B) = p(1-p) = (1 - \boldsymbol{c}_\omega(\delta)^2)/4$. For an ideal cipher, $\boldsymbol{c}_\omega(\delta) \approx 0$ (hence $p \approx 1/2$), while for a real cipher, $\boldsymbol{c}_\omega(\delta)$ may be non-negligible.

Let T_0 count the number of pairs in N samples satisfying $\omega \cdot (C_1 \oplus C_2) = 0$. Then $T_0 \sim \mathrm{Binomial}(N, p)$. By the central limit theorem (for sufficiently large N), T_0 is approximately normally distributed with mean $N \cdot p$ and variance $N \cdot p \cdot (1-p)$. Define $T = (T_0 - T_1)/N = 2T_0/N - 1$ as the empirical estimate of the correlation, where $T_1 = N - T_0$. Then:

$$T_{\mathrm{real}} \sim \mathcal{N}\left(\boldsymbol{c}_\omega(\delta), \frac{1 - \boldsymbol{c}_\omega(\delta)^2}{N}\right), \quad T_{\mathrm{ideal}} \sim \mathcal{N}\left(0, \frac{1}{N}\right). \tag{10}$$

Since B takes values in $\{0, 1\}$, the capacity between the real and ideal distributions can be computed using Definition 5: $\mathrm{Cp}(p_{\mathrm{real}}, p_{\mathrm{ideal}}) = \boldsymbol{c}_\omega(\delta)^2$. By Theorem 3 with $D_{\mathrm{KL}}(p \parallel q) \approx \boldsymbol{c}_\omega(\delta)^2/2$ (by Lemma 7), the sample complexity for distinguishing with type I error α and type II error β is

$$N \approx \frac{(z_\alpha - z_\beta)^2}{\boldsymbol{c}_\omega(\delta)^2}. \tag{11}$$

Multidimensional DL Distinguisher. When multiple DL distinguishers share the same input difference δ but have different output masks forming a linear space of dimension m, we combine them into a multidimensional distinguisher. Let $\omega_1, \ldots, \omega_m$ be a basis for this space and define $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ by

$$F(x) = \left(\omega_1 \cdot (E_k(x) \oplus E_k(x \oplus \delta)), \ldots, \omega_m \cdot (E_k(x) \oplus E_k(x \oplus \delta))\right).$$

By Corollary 1, the sample complexity to distinguish $F(X)$ from uniform over $\mathbb{F}_2^m$ is

$$N \approx \frac{(z_\alpha - z_\beta)^2}{\sum_{\omega \in \mathbb{F}_2^m \setminus \{0\}} \mathrm{Cr}^2(\delta, \omega)}. \tag{12}$$

If all correlations $\boldsymbol{c}_\omega(\delta)$ for $\omega \in \mathbb{F}_2^m \setminus \{0\}$ are approximately equal and non-negligible, the data complexity is reduced by a factor of roughly 2^m compared to a single DL distinguisher.

3 Searching for Differential-Linear Attacks

In this subsection, we present our method to find DL distinguishers. We adapt the approach introduced in CRYPTO 2024 [20] to ciphers with sum-of-permutations (SoP) structures.

Before presenting the details of the distinguisher discovery method, we recall the definitions of DLCT and t-DLCT from [20], as we use them to identify suitable distinguishers and estimate their correlations.

Definition 8 (Differential-Linear Connectivity Table (DLCT) [5,20]). *For a vectorial Boolean function $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$, the DLCT of S is a $2^n \times 2^m$ table whose rows correspond to the input difference δ to S and whose columns correspond to the output mask ω of S. The entry at index (δ, ω) is*

$$\mathit{DLCT}(\delta, \omega) = |\mathit{DLCT}_0(\delta, \omega)| - |\mathit{DLCT}_1(\delta, \omega)|,$$

where $\mathit{DLCT}_b(\delta, \omega) = \{x \in \mathbb{F}_2^n : \ \omega \cdot S(x) \oplus \omega \cdot S(x \oplus \delta) = b\}$.

Sometimes we use the *normalized* DLCT, which is defined as $\overline{\mathtt{DLCT}}(\delta, \omega) \coloneqq 2^{-n} \cdot \mathtt{DLCT}(\delta, \omega)$, and is equal to $\mathrm{Cr}(\omega \cdot S(x) \oplus \omega \cdot S(x \oplus \delta))$. Since the output masks in two sides of DL distinguishers do not need to be the same, one can generalize the definition of DLCT to GDLCT as follows.

Definition 9 (Generalized DLCT (GDLCT)). *For a vectorial Boolean function $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$, the GDLCT of S is a $2^n \times 2^m \times 2^m$ table whose rows correspond to the input difference δ to S and whose columns correspond to the output masks ω_1 and ω_2 of S. The entry at index $(\delta, \omega_1, \omega_2)$ is*

$$\mathit{GDLCT}(\delta, \omega_1, \omega_2) = |\mathit{GDLCT}_0(\delta, \omega_1, \omega_2)| - |\mathit{GDLCT}_1(\delta, \omega_1, \omega_2)|,$$

where $\mathit{GDLCT}_b(\delta, \omega_1, \omega_2) = \{x \in \mathbb{F}_2^n : \ \omega_1 \cdot S(x) \oplus \omega_2 \cdot S(x \oplus \delta) = b\}$.

Definition 10 (Differential-Linear Uniformity [20]). *The differential-linear uniformity of an S-box $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$ is defined as $\mathcal{DLU}(S) \coloneqq \max_{\delta, \omega \neq 0} |\mathit{DLCT}(\delta, \omega)|$.*

Definition 11 (t-DLCT [20]). *For a vectorial Boolean function $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$, the t-DLCT of S is a $2^n \times 2^m$ table such that the entry at index (δ, ω) is*

$$t\text{-}\mathit{DLCT}(\delta, \omega) = \sum_{\delta_{\mathrm{M}} \in \mathbb{F}_2^n} \mathit{DDT}(\delta, \delta_{\mathrm{M}}) \ \cdot \ (t-1)\text{-}\mathit{DLCT}(\delta_{\mathrm{M}}, \omega)\,. \tag{13}$$

For example, the 4-DLCT also referred to as quadruple DLCT (QDLCT) is defined as

$$\texttt{QDLCT}(\delta,\omega) = \sum_{\delta_1,\delta_2,\delta_3 \in \mathbb{F}_2^n} \texttt{DDT}(\delta,\delta_1)\cdot\texttt{DDT}(\delta_1,\delta_2)\cdot\texttt{DDT}(\delta_2,\delta_3)\cdot\texttt{DLCT}(\delta_3,\omega)\,. \quad (14)$$

The DLCT and QDLCT of Orthros S-box are shown in Table 6 and Table 7.

3.1 Searching for Differential-Linear Distinguishers for Each Branch

We first explain how to search for DL distinguishers for each branch of Orthros separately, then describe how to combine the models to find a DL attack for the whole Orthros PRF.

For a block cipher E, we create a CP/MILP model to search for DL distinguishers using the sandwich decomposition $E = E_\ell \circ E_{\mathrm{M}} \circ E_{\mathrm{U}}$ (see Figure 3). We model probabilistic differential propagation through E_{U} and probabilistic linear propagation through E_ℓ to find regular differential and linear trails. For E_{M}, we model deterministic propagation of both differences and linear masks to capture the overlap between differential and linear propagations, measured by the number of shared active S-boxes (or bits).

Finally, we combine all these sub-models into a unified CP/MILP model, connect the junctions, and set the objective function to minimize the weighted sum of: the differential trail weight through E_{U}, the overlap weight through E_{M}, and the linear trail weight through E_ℓ.

The weights are adjusted based on the differential uniformity ($\mathcal{DU}$), differential-linear uniformity ($\mathcal{DLU}$), and linearity ($\mathcal{L}$) of the S-boxes. For the Orthros S-box, $\mathcal{DU} = 2^2$, $\mathcal{DLU} = 2^4$, and $\mathcal{L} = 2^3$. Denoting the number of active S-boxes in the differential trail, the middle overlap, and the linear trail as w_{U}, w_{M}, and w_ℓ respectively, we choose $(w_{\mathrm{U}}, w_{\mathrm{M}}, w_\ell) = (2, 4, 3)$ to scale the contributions appropriately. For more details, see [20].

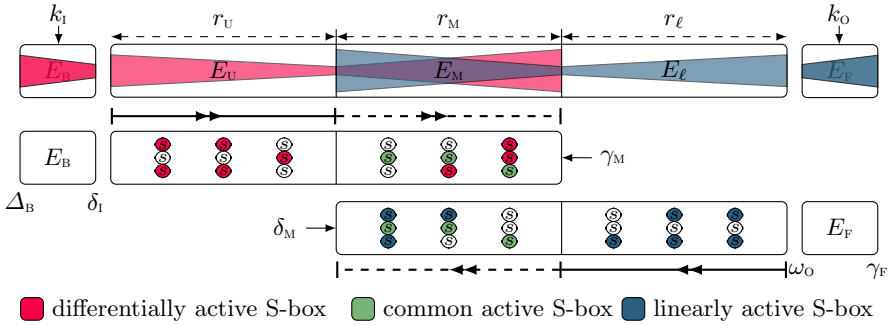


Fig. 3: Searching for complete DL attacks.

The CP/MILP model for the outer parts E_{U} and E_ℓ is a standard CP/MILP model for differential and linear trails [1,40,2]. However, the CP model for the

middle part, i.e., E_{M}, models deterministic trails. In this model, we encode the difference value (resp. linear mask) at each bit using three symbols: 0, 1, and "?", where "?" denotes unknown bits, equivalently, a free bit with respect to the difference value (resp. linear mask). These three symbols are internally encoded as 0, 1, and -1 in the CP model, respectively.

Since CP constraints are not directly supported by some MILP solvers, we also created an alternative pure MILP model for the deterministic propagation of differential and linear trails. In this encoding, each bit position is represented by two binary variables: $(0, 0)$ encodes 0, $(0, 1)$ encodes 1, and $(1, 0)$ encodes "?". We have implemented both approaches in S-box Analyzer tool [21], which now supports generating SAT, MILP, or CP constraints for deterministic propagation (see Section C for details).

We have applied our tool to search for DL distinguishers on `Branch1` and `Branch2` of Orthros. Table 10 and Table 11 summarize the distinguishers we discovered for `Branch1` and `Branch2`, respectively. Since the round function of Orthros varies across different rounds, we define the parameter `Offset` to indicate the starting round of the distinguisher. As shown in Figure 21 and Figure 25, we discovered deterministic DL distinguishers covering up to 5 rounds for both branches, which we experimentally verified.

Prior to our work, the best known DL distinguishers for `Branch1` and `Branch2` of Orthros were the 6-round DL distinguishers proposed in [33]. However, using our method, we found strong DL distinguishers covering up to 10 rounds for each branch. The main reason we can find longer distinguishers is that we consider the dependency between the differential and linear trails across multiple rounds, whereas the method in [33] only considers the dependency at a single S-box layer.

3.2 Searching for Full Differential-Linear Attack

As shown in Figure 3, we extend the CP/MILP model of the distinguisher to also consider critical parameters for key recovery, such as the involved state cells or bits and the corresponding key bits. To achieve this, we propagate the difference backward and the linear mask forward through the key recovery parts E_{B} and E_{F}, respectively.

To search for DL attacks on SoP designs, we modify the model to incorporate constraints specific to this structure. We construct identical CP/MILP sub-models to search for DL distinguishers for each branch. However, due to the output XOR in SoP designs, we must enforce that the output linear masks for both branches are identical. Let $\boldsymbol{c}_1 = 2^{-C_1}$ and $\boldsymbol{c}_2 = 2^{-C_2}$ denote the correlations of the distinguishers for `Branch1` and `Branch2`, respectively. Assuming that the total correlation is approximately $\boldsymbol{c}_t = \boldsymbol{c}_1 \cdot \boldsymbol{c}_2$, we combine the two sub-models into a unified model and set the objective to minimize $C_1 + C_2$.

As illustrated in Figure 4, differential-based attacks on SoP structures typically cannot target the cipher from the end of the distinguisher. Instead, key recovery must be performed before the distinguisher. Therefore, when searching

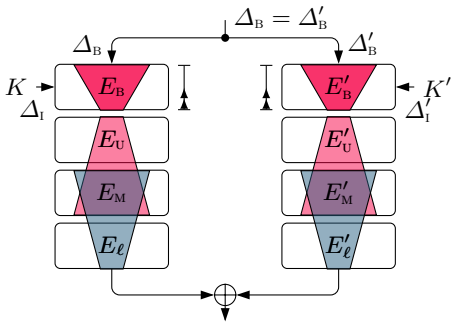


Fig. 4: Full attack setup on SoP designs.

for full attacks on SoP designs, we impose additional constraints on the input differences Δ_B and Δ'_B: they must follow the same activeness pattern. However, the input differences to the distinguisher part, Δ_I and Δ'_I for `Branch1` and `Branch2`, respectively, can take different values.

We applied our model to find full DL attacks on Orthros-PRF. Section H describes the underlying distinguishers and key-recovery structures used in our 6- and 8-round attacks, and Table 12 and Table 13 summarize the distinguishers used in these attacks.

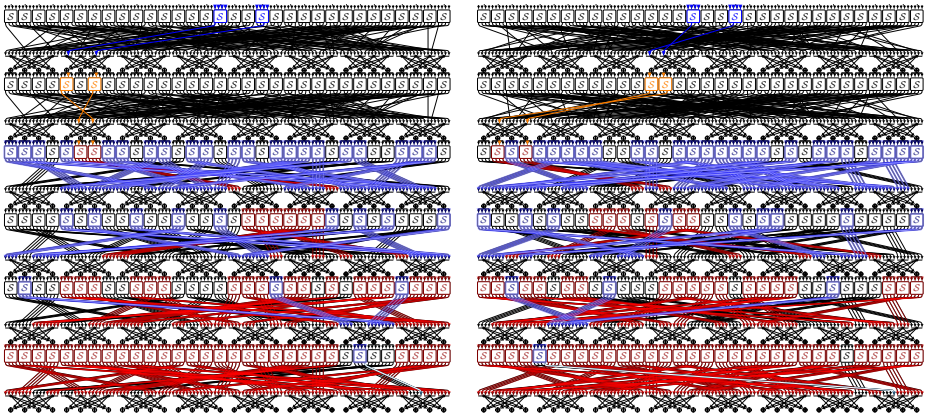
For example, Figure 5 illustrates the structure of our 6-round attack on Orthros-PRF with `Offset` = 2. This structure includes one round of key recovery at the top (involved S-boxes and key bits highlighted in blue). The distinguisher part consists of $1 + 4$ rounds: the first round is a pure differential distinguisher, while the remaining 4 rounds are covered by a combined DL distinguisher. For the combined part, we show the deterministic differential and linear trail propagations together to highlight their overlap. As illustrated in Figure 5, the overlap through the last 4 rounds is zero. Therefore, due to the bit-wise switching effect [20], the correlation over these 4 rounds is expected to be 1, which we experimentally verified.

For all of our DL distinguishers, we experimentally verified the middle part (i.e., E_M) and for most of them provided analytical estimates for the correlation using the generalized `DLCT` framework from [20]. These estimates closely align with the experimental results. For example, the correlation of the middle part in our 7-round DL distinguisher shown in Figure 37 (Table 13) can be expressed as follows for all $\omega \in \mathbb{F}_2^4$:

$$\mathrm{Cr}_{\texttt{Branch1}}(\delta_4, \omega_8) = \frac{\texttt{QDLCT[0x4]}[\omega]}{2^{16}}, \quad \mathrm{Cr}_{\texttt{Branch2}}(\delta'_4, \omega'_8) = \frac{\texttt{QDLCT[0x8]}[\omega]}{2^{16}}, \quad (15)$$

4 Distinguishing Weak vs. Strong Key

Having established key-dependent distinguishers for Orthros-PRF, we now show how to exploit them to reduce key entropy for almost any key. We first address



(a) Branch1 of 6-round Orthros (b) Branch2 of 6-round Orthros

Fig. 5: 6-Round DL attack 0 on Orthros

the problem of classifying whether a given key is weak or strong relative to a specific weak-key distinguisher.

4.1 Problem Formulation

Given a weak-key DL distinguisher, our goal is to determine whether an unknown secret key is weak or strong. Unlike standard distinguishing attacks that distinguish a cipher from a random permutation, we assume the attacker knows the cipher specification (e.g., Orthros) and seeks only to classify the key.

For a weak-key, the distinguisher exhibits bias on certain input assignments (good pairs), while for a strong-key, no exploitable bias exists and outputs are indistinguishable from uniform. We formulate this as a binary hypothesis test:

- H_1 (Weak Key): The key belongs to some weak-key class $\mathcal{W}_i$, and inputs from the corresponding set $\mathcal{X}_i$ produce biased outputs.
- H_0 (Strong Key): The key is strong, and outputs are indistinguishable from uniform regardless of input assignments.

We demonstrate our framework using the 6-round DL attack on Orthros shown in Figure 5. Let m_{in} denote the number of active nibbles in the plaintext (for example, $m_{\text{in}} = 2$), and let b denote the nibble size (for example, $b = 4$). For each weak-key class $\mathcal{W}_i$, let $\mathcal{X}_i$ denote the set of good input pair assignments over the active nibbles, that is, tuples $((v_1, v'_1), (v_2, v'_2), \ldots, (v_{m_{\text{in}}}, v'_{m_{\text{in}}}))$ representing pairs of assignments that satisfy the input difference condition of the DL distinguisher.

4.2 Statistical Framework

In what follows, we assume the correlations $\mathrm{Cr}(a \cdot F)$ (including their signs and magnitudes) for each weak-key class are known from the distinguisher discov-

ery phase described in Section 3.1. These values are verified experimentally or estimated using the generalized DLCT framework [20], as detailed in Section 3.2.

Log-likelihood ratio (LLR) test. We employ a multidimensional DL distinguisher with output dimension m bits. For the 5-round distinguisher in Figure 5, the output mask has one active nibble before MixColumns, giving $m = 4$. All 15 nonzero linear masks yield the same correlation (see Table 12), which we exploit simultaneously.

For a weak-key with distribution $p_y = \Pr(F(X) = y)$ and a strong-key with uniform distribution $q_y = 2^{-m}$, we use the histogram form of the LLR statistic from Equation 6. With N samples and empirical distribution $\widehat{p}_N(y)$, this gives:

$$\mathrm{LLR}_N = N \sum_{y \in \mathbb{F}_2^m} \widehat{p}_N(y) \log \frac{p_y}{q_y}. \tag{16}$$

By Lemma 1, the probabilities p_y are determined by correlations $\mathrm{Cr}(a \cdot F)$ via:

$$p_y = 2^{-m} + \epsilon_y, \quad \text{where} \quad \epsilon_y = 2^{-m} \sum_{a \in \mathbb{F}_2^m \setminus \{0\}} (-1)^{a \cdot y} \mathrm{Cr}(a \cdot F). \tag{17}$$

Since $|\epsilon_y| \ll 2^{-m}$, we use the approximation $\log(1 + x) \approx x$ to simplify:

$$\mathrm{LLR}_N \approx N \sum_{y \in \mathbb{F}_2^m} \widehat{p}_N(y) \frac{\epsilon_y}{2^{-m}} = 2^m N \sum_{y \in \mathbb{F}_2^m} \widehat{p}_N(y)\, \epsilon_y. \tag{18}$$

Data complexity. The required number of samples depends on the capacity $\mathrm{Cp}(F(X))$, which by Theorem 1 equals:

$$\mathrm{Cp}(F(X)) = 2^m \sum_{y \in \mathbb{F}_2^m} \epsilon_y^2 = \sum_{a \in \mathbb{F}_2^m \setminus \{0\}} \mathrm{Cr}^2(a \cdot F). \tag{19}$$

For close distributions, Equation 8 gives $N \approx d/\mathrm{Cp}(F(X))$, where $d = (z_\alpha - z_\beta)^2$ depends on desired error rates.

Threshold determination and error probabilities. By Theorem 2, for a single weak-key class with N samples, the LLR statistic is asymptotically normal as $N \to \infty$:

- Under H_1 (weak-key): $\mathrm{LLR}_N \sim \mathcal{N}(N\mu_0, N\sigma_0^2)$, where $\mu_0 = D_{\mathrm{KL}}(p \parallel q)$ and $\sigma_0^2 = \sum_y p_y \left(\log \frac{p_y}{q_y}\right)^2 - \mu_0^2$.
- Under H_0 (strong-key): $\mathrm{LLR}_N \sim \mathcal{N}(N\mu_1, N\sigma_1^2)$, where $\mu_1 = -D_{\mathrm{KL}}(q \parallel p)$ and $\sigma_1^2 = \sum_y q_y \left(\log \frac{q_y}{p_y}\right)^2 - \mu_1^2$.

For practical computation with close distributions (i.e., when $|p(y) - q(y)| \ll q(y)$ for all y), we use the approximations $\mu_0 \approx \mathrm{Cp}(p, q)/2$, $\mu_1 \approx -\mathrm{Cp}(p, q)/2$, and $\sigma_0^2 \approx \sigma_1^2 \approx \mathrm{Cp}(p, q)$ from Theorem 2, which simplify implementation.

The threshold τ determines the decision rule: reject H_0 (classify as weak) if $\mathrm{LLR}_N \geq \tau$. Following the hypothesis testing framework from Lemma 2, we have:

– **Type II error** (false negative): The probability of failing to detect a weak-key is

$$\beta = \Pr(\mathrm{LLR}_N < \tau \mid H_1) = \Phi\left(\frac{\tau - N\mu_0}{\sqrt{N}\sigma_0}\right). \tag{20}$$

The **detection probability** (power of the test) is $P_{\text{detect}} = 1 - \beta$. Inverting this relation, we obtain the threshold:

$$\tau = N\mu_0 + \sqrt{N}\sigma_0 \cdot \Phi^{-1}(\beta), \tag{21}$$

where Φ is the standard normal CDF.

– **Type I error** (false positive): The probability of incorrectly classifying a strong-key as weak is

$$P_{\text{FP}} = \alpha = \Pr(\mathrm{LLR}_N \geq \tau \mid H_0) = 1 - \Phi\left(\frac{\tau - N\mu_1}{\sqrt{N}\sigma_1}\right). \tag{22}$$

By choosing τ via Equation 21, we control the detection probability P_{detect} (equivalently, the false negative rate $\beta = 1 - P_{\text{detect}}$), and the corresponding false positive probability P_{FP} follows from Equation 22.

The Classification Procedure Our strategy is to test the secret key against all ℓ weak-key classes. If the key is weak, at least one test should detect it; if strong, detections arise only from random false positives.

Step 1: Data collection. Choose N (samples per assignment) based on the capacity $\mathrm{Cp}(F(X)) = \sum_{a \in \mathbb{F}_2^m \setminus \{0\}} \mathrm{Cr}^2(a \cdot F)$ and target detection probability P_{detect}. A larger N improves statistical separation between weak- and strong-keys, reducing both Type II error (missing weak-keys) and the expected number of false positives for strong-keys. The heuristic $N \approx d/\mathrm{Cp}(F(X))$ from Equation 8 provides a starting point, where larger d yields better separation (lower error rates); in practice, N is chosen to achieve negligible expected false positives (see Step 3). For each of the $2^{b \cdot m_{\text{in}}}$ possible active nibble values, prepare N input pairs by varying non-active nibbles. Query $D_{\text{total}} = N \times 2^{b \cdot m_{\text{in}}}$ input pairs total and store all plaintext-ciphertext pairs. This single dataset is reused for testing all classes.

Step 2: Test each weak-key class. For each class $\mathcal{W}_i$ $(i = 1, \ldots, \ell)$:

1. Extract input pairs where active nibbles match values in $\mathcal{X}_i$, giving $M_i = |\mathcal{X}_i| \times N$ pairs.
2. Compute empirical distribution $\widehat{p}_{M_i}(y) = \frac{\mathrm{count}(y)}{M_i}$ for all $y \in \mathbb{F}_2^m$.
3. Compute LLR_{M_i} via Equation 16 and threshold τ_i via Equation 21.
4. If $\mathrm{LLR}_{M_i} \geq \tau_i$, increment detection counter h.

Algorithm 2: Weak-Key vs. Strong-Key Classification

Input: Encryption oracle E_K, input difference δ, output masks $\{\omega_1, \ldots, \omega_m\}$, weak-key classes $\{\mathcal{W}_1, \ldots, \mathcal{W}_\ell\}$ with good assignments $\{\mathcal{X}_1, \ldots, \mathcal{X}_\ell\}$, target detection probability P_{detect}, samples per assignment N

Output: Classification: `WEAK` or `STRONG`

`// Define the vectorial function` $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$

For plaintext pair $(P, P \oplus \delta)$, let $C = E_K(P)$ and $C' = E_K(P \oplus \delta)$, then $F(P) = (\omega_1 \cdot (C \oplus C'), \ldots, \omega_m \cdot (C \oplus C'))$;

`// Step 1: Data collection`

Query $D_{\text{total}} = N \times 2^{b \cdot m_{\text{in}}}$ plaintexts and store plaintext-ciphertext pairs;

`// Precompute bias values (from known correlations)`

Compute $\epsilon_y \leftarrow 2^{-m} \sum_{a \in \mathbb{F}_2^m \setminus \{0\}} (-1)^{a \cdot y} \text{Cr}(a \cdot F)$ for all $y \in \mathbb{F}_2^m$;

`// Precompute expected false positives (analytical)`

$E_{\text{FP}} \leftarrow \sum_{i=1}^{\ell} P_{\text{FP},i}$ where $P_{\text{FP},i} = 1 - \Phi\left(\frac{\tau_i - M_i\mu_1}{\sqrt{M_i}\sigma_1}\right)$;

`// Step 2: Test each weak-key class`

$h \leftarrow 0$; `// number of tests exceeding threshold`

forall *weak-key class* $i = 1, \ldots, \ell$ **do**

  `// Extract pairs` $(P, P \oplus \delta)$ `where active nibbles match` $\mathcal{X}_i$

  $\mathcal{D}_i \leftarrow \{(P, P \oplus \delta) : \text{active nibbles of } (P, P \oplus \delta) \text{ match values in } \mathcal{X}_i\}$;

  $M_i \leftarrow |\mathcal{D}_i|$; `// number of input pairs for test` i

  `// Count occurrences and compute LLR using Equation 18`

  Initialize $n_y \leftarrow 0$ for all $y \in \mathbb{F}_2^m$;

  foreach $(P, P') \in \mathcal{D}_i$ **do**

    Compute $y \leftarrow F(P)$;

    $n_y \leftarrow n_y + 1$;

  $\text{LLR}_{M_i} \leftarrow 2^m \sum_{y \in \mathbb{F}_2^m} n_y \cdot \epsilon_y$;

  $\tau_i \leftarrow M_i\mu_0 + \sqrt{M_i}\sigma_0 \cdot \Phi^{-1}(\beta)$;

  `// Test decision`

  if $\text{LLR}_{M_i} \geq \tau_i$ **then**

    $h \leftarrow h + 1$;

`// Step 3: Final classification`

if $h \geq \theta$ **then**

  return `WEAK` ; `// at least` θ `detections`

else

  return `STRONG` ; `// less than` θ `detections`

Step 3: Final classification. Let h denote the number of tests (among ℓ weak-key classes) for which the LLR exceeds the threshold. Each test i is designed with a target detection probability P_{detect} (power of the test), which determines the threshold τ_i via Equation 21. The corresponding Type I error probability (false positive rate) $P_{\text{FP},i}$ is computed via Equation 22.

For a strong-key (null hypothesis H_0), each test i has false positive probability $P_{\text{FP},i}$. Regardless of test independence, by linearity of expectation, the expected

number of detections is

$$E_{\mathrm{FP}} = \sum_{i=1}^{\ell} P_{\mathrm{FP},i}.$$

If the tests are (approximately) independent, then h is (approximately) Poisson–Binomial with parameters $(P_{\mathrm{FP},1}, \ldots, P_{\mathrm{FP},\ell})$; when $P_{\mathrm{FP},i} \ll 1$ and dependencies are weak, a Poisson(E_{FP}) approximation is often accurate for tail probabilities.

The final classification threshold θ should be chosen to balance Type I error (false positives) and Type II error (false negatives):

- **Conservative threshold:** $\theta = \lceil E_{\mathrm{FP}} + \sqrt{E_{\mathrm{FP}}} \rceil$ minimizes Type I error, classifying as weak only when h significantly exceeds the expected noise level E_{FP}.
- **Balanced threshold:** $\theta = \lceil E_{\mathrm{FP}} \rceil$ provides a middle ground, suitable when E_{FP} is moderate.
- **Aggressive threshold:** $\theta = 1$ classifies as weak if *any* test detects, maximizing detection power (minimizing Type II error) at the cost of higher Type I error when $E_{\mathrm{FP}} \gg 1$.

The choice depends on the application requirements. When the cost of misclassifying a strong-key (Type I error) is high, a conservative threshold is preferable. When the cost of missing a weak-key (Type II error) is high, an aggressive threshold is preferable. In our experiments, we report results for an aggressive $\theta = 1$ to illustrate detection power at small N; for claims requiring controlled Type I error, one must take θ according to E_{FP} (see Section 20).

Handling class intersections. When the key belongs to weak-key class $\mathcal{W}_j$, testing a different class $\mathcal{W}_i$ with overlapping good assignments ($\mathcal{X}_i \cap \mathcal{X}_j \neq \emptyset$) may still trigger detection. The shared assignments in $\mathcal{X}_i \cap \mathcal{X}_j$ exhibit the bias characteristic of $\mathcal{W}_j$, though the overall signal is diluted by non-shared assignments in $\mathcal{X}_i \setminus \mathcal{X}_j$. Crucially, since we test *all* ℓ classes, we always test the correct class $\mathcal{W}_j$ with full detection power P_{detect}. Overlapping tests provide redundant detection opportunities without increasing the false positive rate on strong-keys. Thus, the net effect is positive: testing all classes ensures robust detection of any weak-key while maintaining controlled Type I error.

Alternative methodology. An alternative classification approach based on the maximum statistic with rigorous FWER control is presented in Section D. However, our LLR-based method achieves 62–120× better data efficiency for the Orthros attacks.

4.3 Experimental Verification

We implemented the attack against 6-round Orthros to verify our theoretical analysis. We used the DL distinguisher shown in Figure 5. The output mask has

one active nibble before MixColumns, and any nonzero mask yields an autocorrelation of approximately $2^{-5} \times 2^{-6} = 2^{-11}$. By Equation 19, the capacity is $15 \times 2^{-22} \approx 2^{-18.09}$.

We set $N = 2^{20}$, giving data complexity $2^8 \times N = 2^{28}$ plaintext-ciphertext pairs. For each weak-key class test, we choose threshold τ_i to achieve target detection probability $P_{\text{detect}} = 0.7$. This yields an expected number of false detections $E_{\text{FP}} = \sum_i P_{\text{FP},i} \approx 8.19$ for strong-keys.

We first verified the detection probability by randomly selecting keys from a weak-key class. Over 1000 trials, we correctly identified the weak-key class 750 times, achieving empirical success probability 0.75. This slightly exceeds the theoretical value of 0.70, indicating our model conservatively estimates detection power. Next, we verified E_{FP} by randomly selecting strong-keys. Over 1000 trials, we observed an average of 8.00 false detections per strong-key, closely matching the theoretical estimate of 8.19. The theoretical model accurately predicts the false positive rate under the null hypothesis.

Example decisions. Using an aggressive threshold $\theta = 1$ (classify as weak if $h \geq 1$):

- $h = 0$: classify as **strong-key** (no detections).
- $h \geq 1$: classify as **weak-key** (at least one detection).

With $E_{\text{FP}} \approx 8.19$, strong-keys trigger at least one false detection with probability $P(h \geq 1 \mid H_0) \approx 1 - e^{-E_{\text{FP}}} \approx 0.9997$ (Poisson approximation), so $\theta = 1$ is unsuitable when $E_{\text{FP}} \gg 1$. For practical classification at this N, use the balanced threshold $\theta = \lceil E_{\text{FP}} \rceil = 9$ or a conservative choice $\theta = \lceil E_{\text{FP}} + \sqrt{E_{\text{FP}}} \rceil = 12$; the corresponding tail probabilities are summarized in Table 3.

Controlling error rates via parameter tuning. The expected number of false positives $E_{\text{FP}} = \sum_i P_{\text{FP},i}$ can be reduced by adjusting two parameters:

- **Increasing N:** More plaintext-ciphertext pairs improve statistical separation between H_0 and H_1. For each test i, we have $M_i = |\mathcal{X}_i| \cdot N$ samples. The Type I error $P_{\text{FP},i} = 1 - \Phi\left(\frac{\tau_i - M_i\mu_1}{\sqrt{M_i}\sigma_1}\right)$ has the form $1 - \Phi(a\sqrt{N} - b)$ for positive constants a, b depending on the capacity and success probability. By Lemma 3, this decreases exponentially in N. For example, the full 6-round attack (Table 4) uses $N = 2^{32}$ (versus $N = 2^{20}$ here), achieving $E_{\text{FP}} \approx 0$ compared to $E_{\text{FP}} \approx 8.19$ here.
- **Decreasing P_{detect}:** Lower target detection probability decreases $\Phi^{-1}(P_{\text{detect}})$, which raises the threshold $\tau_i = M_i\mu_0 - \sqrt{M_i}\sigma_0 \cdot \Phi^{-1}(P_{\text{detect}})$. A higher threshold reduces both the detection power for weak-keys and the false positive rate $P_{\text{FP},i}$ for strong-keys. However, we do not pursue this option as it reduces detection power, and prefer to increase N instead.

The experimental setup here ($N = 2^{20}$, $P_{\text{detect}} = 0.7$) prioritizes demonstration clarity over error minimization. In practical attacks requiring low false positive rates, we increase N until $E_{\text{FP}} \ll 1$, enabling reliable classification with threshold

Table 3: Type I error decay with data complexity for 6-round attack (Cp = $2^{-18.0931}$, $\ell = 2240$, $P_{\text{detect}} = 0.7$)

N	E_{FP}	$P(h \geq 1 \mid H_0)$	Balanced θ	Conservative θ
2^{18}	212.4992	1.0000	213	228
2^{19}	61.1348	1.0000	62	69
2^{20}	8.1905	0.9997	9	12
2^{21}	0.2344	0.2089	1	1
2^{22}	0.0002	0.0002	1	1
2^{23}	0.0000	0.0000	1	1

$\theta = 1$. Table 3 demonstrates the exponential decay of the expected false positive count E as N increases for the 6-round attack (Distinguisher 0, capacity $2^{-18.09}$, $\ell = 2240$ weak-key classes). With $N = 2^{20}$, we achieve $E_{\text{FP}} \approx 8.2$, indicating substantial expected noise. For $N \geq 2^{22}$ and beyond, E_{FP} becomes negligible (≤ 0.0002).

More empirical observations. Extensive experiments with randomly selected keys for 4-round (Figure 14) and 5-round (Figure 5) distinguishers are shown in Figure 15–Figure 20 and Section E. Weak keys exhibit clustered correlation peaks at class-specific nibble assignments, while strong-keys follow half-normal baseline noise.

5 Combining Multiple Weak-Key Distinguishers

Having classified keys as weak or strong for individual distinguishers, we now aggregate information from multiple weak-key distinguishers to quantify the total entropy reduction achieved. A single weak-key distinguisher always leaves a relatively large fraction of keys undetected. However, there are often multiple weak-key distinguishers, each targeting different key bits. Thus, by combining them, we can extract more comprehensive information about the secret key.

The statistical problem. When we combine multiple distinguishers that target different key bits, we face a dependent multiple testing problem. Let J be the number of distinguishers. For distinguisher $j \in \{1, \ldots, J\}$, let ℓ_j be its number of weak-key classes, and index classes by $i \in \{1, \ldots, \ell_j\}$. The total number of hypothesis tests is $\ell = \sum_{j=1}^{J} \ell_j$. For each class (j, i) we test H_0 (the key is strong for class (j, i)) versus H_1 (the key belongs to weak-key class (j, i)) using the LLR statistic from Section 4. These tests are dependent because distinguishers may operate on overlapping subsets of the key.

Error control across distinguishers. As shown in Section 4, each test (j, i) has false positive probability $P_{\text{FP},j,i}$. If tests operated on disjoint key bits, their information would add. In our setting, many tests share bits, so they are dependent.

By linearity of expectation, the expected number of false positives across all ℓ tests equals $\sum_{j=1}^{J}\sum_{i=1}^{\ell_j} P_{\mathrm{FP},j,i}$.

Definition 12 (Family-Wise Error Rate). *The family-wise error rate (FWER) is the probability of at least one false positive across all tests:*

$$FWER = \Pr\left(\bigcup_{j=1}^{J}\bigcup_{i=1}^{\ell_j}\{\textit{reject } H_0 \textit{ for class } (j,i)\} \;\middle|\; \textit{the key is strong for all classes}\right).$$

Theorem 4 (Bonferroni Bound). *By the union bound, regardless of dependencies among tests,*

$$FWER \le \sum_{j=1}^{J}\sum_{i=1}^{\ell_j} P_{FP,j,i}.$$

Consequently, choosing per-test false positive probabilities such that $\sum_{j,i} P_{FP,j,i} \le \alpha$ guarantees $FWER \le \alpha$.

Remark 1 (Poisson Approximation). When $P_{\mathrm{FP},j,i} \ll 1$ and $\sum_{j,i} P_{\mathrm{FP},j,i} \ll 1$, the number of false detections h is approximately Poisson-distributed, yielding

$$\Pr(h \ge 1 \mid H_0) \approx 1 - \exp\left(-\sum_{j,i} P_{\mathrm{FP},j,i}\right).$$

We use this approximation diagnostically but rely on the Bonferroni bound for rigorous FWER control.

Information gain via weak-key coverage. After running all distinguishers, we obtain J families of weak-key classes $\{\mathcal{W}_{j,i}\}_{i=1}^{\ell_j}$ for $j = 1, \ldots, J$, and define the strong-key set $\mathcal{S} = \mathcal{K} \setminus \left(\cup_{j=1}^{J}\cup_{i=1}^{\ell_j}\mathcal{W}_{j,i}\right)$. For the entropy calculation below, we flatten these into a single enumeration $\mathcal{W}_1, \ldots, \mathcal{W}_\ell$ where $\ell = \sum_{j=1}^{J}\ell_j$. Note that keys may belong to multiple weak-key classes.

Beyond controlling false positives, we wish to quantify how much information each distinguisher provides about the key. When a distinguisher classifies a key into a weak-key class of size $|\mathcal{W}_i|$, it narrows the key space, yielding information gain. To measure the expected information gain over all possible keys, we use Shannon entropy of the partition induced by the distinguishers.

Definition 13 (Shannon Entropy of Key Partition). *Let κ_j denote the number of distinct key bits involved in distinguisher j, and let the distinguisher partition this κ_j-bit subspace into strong-key set $\mathcal{S}$ and weak-key classes $\{\mathcal{W}_i\}_{i=1}^{\ell_j}$. For a uniformly random key drawn from the κ_j-bit subspace, the expected information gain (in bits) is quantified by Shannon entropy:*

$$H_j = -\frac{|\mathcal{S}|}{2^{\kappa_j}}\log_2\frac{|\mathcal{S}|}{2^{\kappa_j}} - \sum_{i=1}^{\ell_j}\frac{|\mathcal{W}_i|}{2^{\kappa_j}}\log_2\frac{|\mathcal{W}_i|}{2^{\kappa_j}}.$$

Remark 2. If distinguishers were independent, total information would equal the sum of individual test entropies. However, our tests overlap (they share key bits), so joint entropy is strictly less than the sum of marginal entropies. We estimate joint entropy empirically by sampling random keys because the dependency structure is too complex for analytic evaluation.

5.1 DL Attack on 6-Round Orthros

We now demonstrate our framework for combining multiple weak-key distinguishers on 6-round Orthros using four DL distinguishers. Table 4 summarizes the key parameters for each distinguisher. We present the attack systematically in six steps.

Step 1: Identify the distinguishers. From Table 4 and Table 12, we have four distinguishers targeting different weak-key classes:

- Distinguisher 0: 2240 weak-key classes, capacity $2^{-18.09}$
- Distinguisher 1: 1920 weak-key classes, capacity $2^{-26.09}$
- Distinguisher 2: 2128 weak-key classes, capacity $2^{-23.90}$
- Distinguisher 3: 3136 weak-key classes, capacity $2^{-27.68}$

Each distinguisher involves two nibbles in different positions of the input for the key recovery.

Step 2: Determine sample size and data complexity. We choose sample size N based on the minimum capacity across all distinguishers. Distinguisher 3 (see Table 12) has the smallest capacity at $2^{-27.68}$. To ensure the expected number of false positives stays well below 1 for this bottleneck distinguisher, we set $N = 2^{32}$.

With this choice, the expected false positives per distinguisher from Table 4 are:

$$\sum_{i=1}^{\ell_0} P_{\mathrm{FP},i}^{(0)} \approx 0, \quad \sum_{i=1}^{\ell_1} P_{\mathrm{FP},i}^{(1)} = 2^{-48.57}, \quad \sum_{i=1}^{\ell_2} P_{\mathrm{FP},i}^{(2)} \approx 0, \quad \sum_{i=1}^{\ell_3} P_{\mathrm{FP},i}^{(3)} = 0.0080.$$

By linearity of expectation, the total expected number of false positives across all four distinguishers is $E_{\mathrm{FP}} = 0.0080 + 2^{-48.57} \approx 0.008 \ll 1$, confirming our choice.

For data complexity, we collect N pairs by varying eight active nibbles (two per distinguisher) plus two additional nibbles, yielding 2^{40} plaintext-ciphertext pairs.

Step 3: Compute time complexity for classification. For each distinguisher, we test each weak-key class by identifying the associated set $\mathcal{X}_i$ and computing the LLR statistic using $|\mathcal{X}_i| \times N$ pairs. The time complexity per distinguisher is $\sum_{i=1}^{\ell_j} |\mathcal{X}_i| \times N$, where ℓ_j is the number of weak-key classes in distinguisher j. From Table 4, $\sum_{i=1}^{\ell_j} |\mathcal{X}_i| \approx 2^{13.27}, 2^{12.81}, 2^{15.85}, 2^{16.61}$ for distinguishers 0, 1, 2, 3 respectively. Combining all four distinguishers gives total classification time:

$$(2^{13.27} + 2^{12.81} + 2^{15.85} + 2^{16.61}) \times 2^{32} \approx 2^{49.43}.$$

Table 4: Summary of DL distinguishers for 6-round attack, where $N = 2^{32}$.

	active nibble index	Δ	key bits	#weak-key classes	#strong keys	capacity (log 2)	entropy	$\sum_i \lvert\mathcal{X}_i\rvert$ (log 2)	$\sum_i P_{\mathrm{FP},i}$
Dist.0 Figure 5	15	0x8	25 21 12 24	2240	52992	-18.09	2.74	13.27	≈ 0
		0x4	42 55 29 50						
	18	0x2	57 70 100 64						
		0x4	20 97 28 32						
Dist.1 Figure 31	20	0x1	11 65 71 59	1920	56576	-26.09	2.01	12.81	$2^{-48.57}$
		0x4	43 2 83 58						
	21	0x2	72 95 106 37						
		0x1	27 117 111 63						
Dist.2 Figure 32	23	0x4	105 30 47 9	2128	24576	-23.9	3.41	15.85	≈ 0
		0x4	92 56 11 41						
	31	0x4	77 32 4 116						
		0x4	94 16 51 4						
Dist.3 Figure 33	5	0x4	1 40 28 67	3136	49152	-27.68	3.56	16.61	0.0080
		0x4	84 108 115 59						
	10	0x4	117 126 99 68						
		0x4	48 61 124 62						

Step 4: Compute Shannon entropy per distinguisher. Using the Shannon entropy formula from the definition above, we compute the expected information gain for each distinguisher. For distinguisher j with κ_j distinct key bits, strong-key set $\mathcal{S}$ and weak-key classes $\{\mathcal{W}_i\}_{i=1}^{\ell_j}$, the entropy is:

$$H_j = -\frac{|\mathcal{S}|}{2^{\kappa_j}} \log_2 \frac{|\mathcal{S}|}{2^{\kappa_j}} - \sum_{i=1}^{\ell_j} \frac{|\mathcal{W}_i|}{2^{\kappa_j}} \log_2 \frac{|\mathcal{W}_i|}{2^{\kappa_j}}.$$

We compute this entropy exactly for each distinguisher; values appear in the "entropy" column of Table 4.

For example, from Table 4, distinguisher 0 involves $\kappa_0 = 16$ distinct key bits and has $|\mathcal{S}| = 52992$ strong-keys out of $2^{16} = 65536$ total (approximately 80.9%), yielding entropy $H_0 = 2.74$ bits. For the typical case where a random key is strong, we learn only $-\log_2(52992/65536) \approx 0.31$ bits (the key avoids all weak-key classes). However, distinguisher 0 contains weak-key classes as small as $|\mathcal{W}_i| = 1$ (singleton classes). A key in such a class reveals all $\kappa_0 = 16$ bits exactly. This demonstrates the power of combining multiple distinguishers: even if most keys yield modest information from any single distinguisher, a key that is weak for at least one distinguisher can provide substantial information gain.

Step 5: Estimate combined information gain experimentally. The four distinguishers touch overlapping key bits, so we cannot simply add their individual information gains. We estimate the combined information using Monte Carlo simulation over 10000 randomly sampled keys.

Recall from Step 1 that each distinguisher has weak-key classes $\mathcal{W}_1, \ldots, \mathcal{W}_\ell$, where keys in class $\mathcal{W}_i$ share the same set of good input assignments $\mathcal{X}_i$. Because distinguishers share key-bit positions, we must account for dependencies. We identify the overlapping bits (positions that appear in multiple distinguishers) and condition on their 2^k possible values. Once we fix the overlapping bits, the distinguishers become conditionally independent because each then reads a disjoint set of key bits.

For each sampled 128-bit key, we estimate how much the four distinguishers narrow down the key space by computing the fraction of possible 16-bit values (per distinguisher) that remain consistent with this key. The algorithm:

1. For each of the 2^k possible values of the overlapping bits:
   (a) For each distinguisher:
      – Extract the key-bit values that are relevant to this distinguisher.
      – Identify which weak-key class the sampled key belongs to.
      – Count how many 16-bit assignments in that class match the fixed overlapping bits.
      – Divide by the total number of 16-bit assignments (across all classes) that match the fixed overlapping bits to get a conditional probability.
   (b) Multiply the four conditional probabilities (valid under conditional independence).
2. Average the products over all 2^k overlapping-bit values to obtain probability p.
3. Record $-\log_2 p$ as the information gain for this key.

The histogram in Figure 6 shows the distribution of information values over the 10000 samples. The vertical lines mark the mean and median.

The mean information gain is 11.65 bits and the median is 12.93 bits. In the worst case, 39.05% of keys are classified as strong by all four distinguishers. Even here, we learn 1.35 bits (the key avoids all weak-key classes), reducing the final exhaustive search cost from 2^{128} to $2^{128-1.35} \approx 2^{126.65}$. In the best case (0.03% of keys), we obtain 52.41 bits.

Step 6: Compute total attack complexity. The total attack cost equals the classification time plus the exhaustive search over the remaining key space. The median indicates that for half of all keys, exhaustive search requires at most $2^{128-12.93} = 2^{115.07}$ operations. Combined with classification time $2^{49.43}$, the total median attack complexity is dominated by exhaustive search at $2^{115.07}$.

5.2 DL Attack on 8-Round Orthros

We now scale our framework to 8-round Orthros using four 7-round DL distinguishers in Table 13. We add one round on the plaintext side for key recovery and face the same dependent multiple testing problem as in the 6-round case. Table 5 summarizes the key parameters for each distinguisher. We present the attack systematically in six steps.

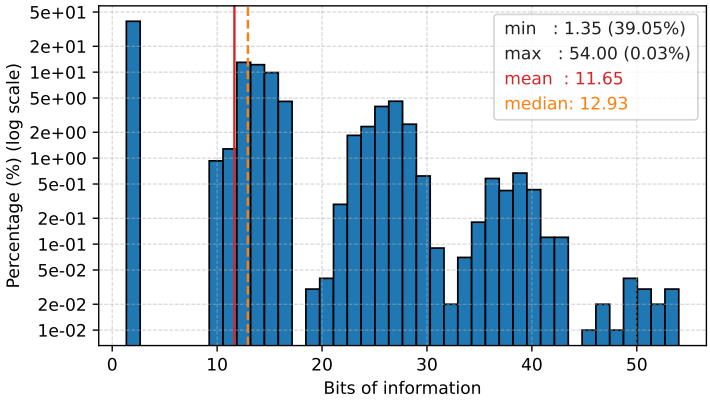


Fig. 6: Histogram of the bit information obtained by four 5-round distinguishers

Step 1: Identify the distinguishers. From Table 5 and Table 13, we have four distinguishers targeting different weak-key classes:

- Distinguisher 0: 800 weak-key classes, capacity $2^{-83.19}$
- Distinguisher 1: 896 weak-key classes, capacity $2^{-91.87}$
- Distinguisher 2: 2688 weak-key classes, capacity $2^{-95.30}$
- Distinguisher 3: 2240 weak-key classes, capacity $2^{-99.64}$

Each distinguisher involves two nibbles in different positions of the input for the key recovery.

Step 2: Determine sample size and data complexity. We choose sample size N based on the minimum capacity across all distinguishers. Distinguisher 3 (see Table 13) has the smallest capacity at $2^{-99.64}$. Using the relation $N \approx d/\text{capacity}$ for target $d \approx 2^{-5}$ (expected false positives well below 1), we set $N = 2^{104}$.

With this choice, the expected false positives per distinguisher from Table 5 are:

$$\sum_{i=1}^{\ell_0} P_{\mathrm{FP},i}^{(0)} \approx 0, \quad \sum_{i=1}^{\ell_1} P_{\mathrm{FP},i}^{(1)} \approx 0, \quad \sum_{i=1}^{\ell_2} P_{\mathrm{FP},i}^{(2)} \approx 0, \quad \sum_{i=1}^{\ell_3} P_{\mathrm{FP},i}^{(3)} = 0.012.$$

By linearity of expectation, the total expected number of false positives across all four distinguishers is $E_{\mathrm{FP}} = 0.012 \ll 1$, confirming our choice.

For data complexity, we collect N pairs by varying eight active nibbles (two per distinguisher) plus 22 additional nibbles, yielding 2^{112} plaintext-ciphertext pairs.

Step 3: Compute time complexity for classification. For each distinguisher, we test each weak-key class by identifying the associated set $\mathcal{X}_i$ and computing the

Table 5: Summary of DL distinguishers for 8-round attack, where $N = 2^{104}$.

	active nibble			#weak-key classes	#strong keys	capacity (log 2)	entropy	$\sum_i \lvert\mathcal{X}_i\rvert$ (log 2)	$\sum_i P_{\mathrm{FP},i}$
	index	Δ	key bits						
Dist.0 Figure 34	27	0x4	10 64 117 21	800	7072	-83.19	1.88	11.54	≈ 0
		0x8	88 84 10 35						
	29	0x4	88 112 78 14						
		0x1	42 78 17 114						
Dist.1 Figure 35	20	0x1	53 93 44 27	896	14144	-91.87	1.88	11.49	≈ 0
		0x4	21 44 29 109						
	21	0x2	90 101 122 2						
		0x1	58 27 69 11						
Dist.2 Figure 36	6	0x4	12 116 115 97	2688	51200	-95.30	3.14	14.81	≈ 0
		0x4	94 123 61 98						
	24	0x2	126 59 91 6						
		0x1	41 16 117 104						
Dist.3 Figure 37	15	0x8	127 8 31 62	2240	52992	-99.64	2.74	13.27	0.012
		0x4	40 28 72 122						
	18	0x2	103 3 99 55						
		0x4	82 125 26 54						

LLR statistic using $|\mathcal{X}_i| \times N$ pairs. The time complexity per distinguisher is $\sum_{i=1}^{\ell_j} |\mathcal{X}_i| \times N$, where ℓ_j is the number of weak-key classes in distinguisher j. From Table 5, $\sum_{i=1}^{\ell_j} |\mathcal{X}_i| \approx 2^{11.54}, 2^{11.49}, 2^{14.81}, 2^{13.27}$ for distinguishers 0, 1, 2, 3 respectively. Combining all four distinguishers gives total classification time:

$$(2^{11.54} + 2^{11.49} + 2^{14.81} + 2^{13.27}) \times 2^{104} \approx 2^{119.44}.$$

Step 4: Compute Shannon entropy per distinguisher. Using the Shannon entropy formula from the definition above, we compute the expected information gain for each distinguisher. For distinguisher j with κ_j distinct key bits, strong-key set $\mathcal{S}$ and weak-key classes $\{\mathcal{W}_i\}_{i=1}^{\ell_j}$, the entropy is:

$$H_j = -\frac{|\mathcal{S}|}{2^{\kappa_j}} \log_2 \frac{|\mathcal{S}|}{2^{\kappa_j}} - \sum_{i=1}^{\ell_j} \frac{|\mathcal{W}_i|}{2^{\kappa_j}} \log_2 \frac{|\mathcal{W}_i|}{2^{\kappa_j}}.$$

We compute this entropy exactly for each distinguisher; values appear in the "entropy" column of Table 5.

Step 5: Estimate combined information gain empirically. The four distinguishers target overlapping key bits, so their information gains are not independent. If tests were independent, we could simply add individual entropies. However, because they share key bits, joint entropy is strictly less than the sum of marginal entropies. Exact evaluation requires tracking complex dependencies among the 16-bit subkeys. Furthermore, information gain varies widely depending on which

weak-key classes the full key satisfies. We therefore estimate the joint distribution empirically by sampling 10,000 random 128-bit keys and applying all four distinguishers simultaneously to each key. To do this, we follow the same algorithm as in the 6-round case.

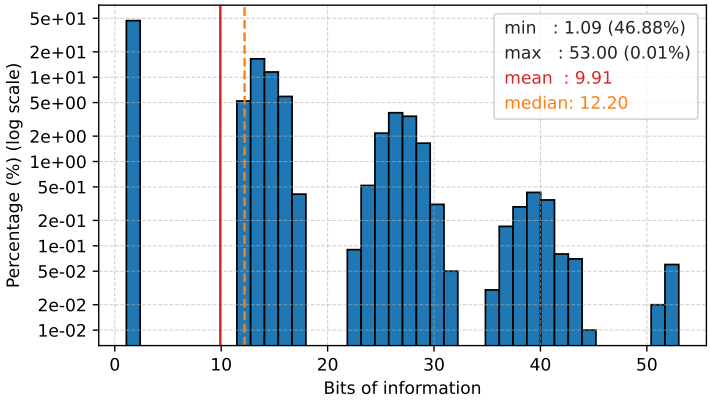


Fig. 7: Histogram of the bit information obtained by four 7-round distinguishers

Figure 7 shows the resulting histogram. The mean information gain is 9.91 bits and the median is 12.20 bits. In the worst case, 46.88% of keys are classified as strong by all four distinguishers. Even here, we learn 1.09 bits (the key avoids all weak-key classes), reducing the final exhaustive search cost from 2^{128} to $2^{128-1.09} \approx 2^{126.91}$.

Step 6: Compute total attack complexity. The total attack cost equals the classification time plus the exhaustive search over the remaining key space. In the worst case, 46.88% of keys are classified as strong by all distinguishers, yielding only 1.09 bits. For these keys, the attack complexity is:

$$2^{119.44} + 2^{128-1.09} \approx 2^{126.92}.$$

For the median case, we obtain 11.87 bits, giving attack complexity:

$$2^{119.44} + 2^{128-11.87} \approx 2^{119.58}.$$

The median attack complexity is $2^{119.58}$, dominated by the classification phase.

6 Conclusion and Future Works

We presented a generic approach for turning key-dependent attacks into universal attacks. As a concrete application, we applied our method to differential-linear attacks. An interesting direction for future work is to explore the same idea in other cryptanalytic settings, such as differential and boomerang attacks.

We focused on Orthros as a representative sum-of-permutations (SoP) design, where weak-key behavior is inherent. Moreover, for Orthros, DL distinguishers are currently the most effective known distinguishers. Beyond this case study, our method is generic and should transfer to other primitives.

Permutation-based constructions are particularly promising targets. They often admit structurally strong-key-dependent distinguishers that can be replicated due to invariant properties of the primitive. For example, rotational invariance in `Ascon` enables the replication of conditional or weak-key DL distinguishers. Applying our framework to such permutation-based primitives is another natural direction for future work.

References

1. Abdelkhalek, A., Sasaki, Y., Todo, Y., Tolba, M., Youssef, A.M.: MILP modeling for (large) S-boxes to optimize probability of differential characteristics. IACR Trans. Symmetric Cryptol. **2017**(4), 99–129 (2017). https://doi.org/10.13154/tosc.v2017.i4.99-129
2. Ankele, R., Kölbl, S.: Mind the gap - A closer look at the security of block ciphers against differential cryptanalysis. In: Cid, C., Jr., M.J.J. (eds.) SAC 2018. LNCS, vol. 11349, pp. 163–190. Springer (2018). https://doi.org/10.1007/978-3-030-10970-7_8
3. Baignères, T., Junod, P., Vaudenay, S.: How far can we go beyond linear cryptanalysis? In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 432–450. Springer (2004). https://doi.org/10.1007/978-3-540-30539-2_31
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. Bar-On, A., Dunkelman, O., Keller, N., Weizman, A.: DLCT: A new tool for differential-linear cryptanalysis. In: EUROCRYPT 2019. LNCS, vol. 11476, pp. 313–342. Springer (2019). https://doi.org/10.1007/978-3-030-17653-2_11
6. Bellare, M., Krovetz, T., Rogaway, P.: Luby-rackoff backwards: Increasing security by making block ciphers non-invertible. In: Nyberg, K. (ed.) EUROCRYPT '98. LNCS, vol. 1403, pp. 266–280. Springer (1998). https://doi.org/10.1007/BFB0054132
7. Ben-Aroya, I., Biham, E.: Differential cryptanalysis of lucifer. J. Cryptol. **9**(1), 21–34 (1996). https://doi.org/10.1007/BF02254790
8. Beyne, T.: Block cipher invariants as eigenvectors of correlation matrices. In: Peyrin, T., Galbraith, S.D. (eds.) ASIACRYPT 2018. LNCS, vol. 11272, pp. 3–31. Springer (2018). https://doi.org/10.1007/978-3-030-03326-2_1
9. Beyne, T.: Block cipher invariants as eigenvectors of correlation matrices. J. Cryptol. **33**(3), 1156–1183 (2020). https://doi.org/10.1007/S00145-020-09344-1
10. Beyne, T., Rijmen, V.: Differential cryptanalysis in the fixed-key model. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022. LNCS, vol. 13509, pp. 687–716. Springer (2022). https://doi.org/10.1007/978-3-031-15982-4_23
11. Blondeau, C., Leander, G., Nyberg, K.: Differential-linear cryptanalysis revisited. J. Cryptol. **30**(3), 859–888 (2017). https://doi.org/10.1007/s00145-016-9237-5

12. Brayton, R.K., Hachtel, G.D., McMullen, C.T., Sangiovanni-Vincentelli, A.L.: Logic Minimization Algorithms for VLSI Synthesis, The Kluwer International Series in Engineering and Computer Science, vol. 2. Springer (1984). https://doi.org/10.1007/978-1-4613-2821-6
13. Cover, T.M., Thomas, J.A.: Elements of information theory (2. ed.). Wiley (2006)
14. Dai, W., Hoang, V.T., Tessaro, S.: Information-theoretic indistinguishability via the chi-squared method. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10403, pp. 497–523. Springer (2017). https://doi.org/10.1007/978-3-319-63697-9_17
15. Dobbertin, H., Bosselaers, A., Preneel, B.: RIPEMD-160: A strengthened version of RIPEMD. In: Gollmann, D. (ed.) FSE 1996. LNCS, vol. 1039, pp. 71–82. Springer (1996). https://doi.org/10.1007/3-540-60865-6_44
16. Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M.: Ascon v1.2: Lightweight authenticated encryption and hashing. Journal of Cryptology **34**(3), 33 (2021). https://doi.org/10.1007/s00145-021-09398-9
17. Dunkelman, O., Keller, N., Shamir, A.: A practical-time related-key attack on the KASUMI cryptosystem used in GSM and 3G telephony. J. Cryptol. **27**(4), 824–849 (2014). https://doi.org/10.1007/s00145-013-9154-9
18. Flórez-Gutiérrez, A., Grassi, L., Leander, G., Sibleyras, F., Todo, Y.: General practical cryptanalysis of the sum of round-reduced block ciphers and ZIP-AES. In: Chung, K., Sasaki, Y. (eds.) ASIACRYPT 2024. LNCS, vol. 15492, pp. 280–311. Springer (2024). https://doi.org/10.1007/978-981-96-0947-5_10
19. Gauravaram, P., Knudsen, L.R., Matusiewicz, K., Mendel, F., Rechberger, C., Schläffer, M., Thomsen, S.S.: Grøstl - a SHA-3 candidate. In: Handschuh, H., Lucks, S., Preneel, B., Rogaway, P. (eds.) Symmetric Cryptography, 11.01. - 16.01.2009. Dagstuhl Seminar Proceedings, vol. 09031. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany (2009), http://drops.dagstuhl.de/opus/volltexte/2009/1955/
20. Hadipour, H., Derbez, P., Eichlseder, M.: Revisiting differential-linear attacks via a boomerang perspective with application to AES, Ascon, CLEFIA, SKINNY, PRESENT, KNOT, TWINE, WARP, LBlock, Simeck, and SERPENT pp. 38–72 (2024). https://doi.org/10.1007/978-3-031-68385-5_2
21. Hadipour, H., Nageler, M., Eichlseder, M.: Throwing boomerangs into feistel structures application to clefia, warp, lblock, lblock-s and TWINE. IACR Trans. Symmetric Cryptol. **2022**(3), 271–302 (2022). https://doi.org/10.46586/tosc.v2022.i3.271-302
22. Hawkes, P.: Differential-linear weak key classes of IDEA. In: Nyberg, K. (ed.) EUROCRYPT '98. LNCS, vol. 1403, pp. 112–126. Springer (1998). https://doi.org/10.1007/BFB0054121
23. Hermelin, M., Cho, J.Y., Nyberg, K.: Multidimensional linear cryptanalysis. J. Cryptol. **32**(1), 1–34 (2019). https://doi.org/10.1007/S00145-018-9308-X
24. Hou, S., Wu, B., Wang, S., Guo, H., Lin, D.: Truncated differential attacks on symmetric primitives with linear key schedule: WARP and orthros. Comput. J. **67**(4), 1483–1500 (2024). https://doi.org/10.1093/COMJNL/BXAD075
25. Jr., B.S.K., Rivest, R.L., Sherman, A.T.: Is DES a pure cipher? (results of more cycling experiments on DES). In: Williams, H.C. (ed.) CRYPTO '85. LNCS, vol. 218, pp. 212–226. Springer (1985). https://doi.org/10.1007/3-540-39799-X_17
26. Kara, O., Manap, C.: A new class of weak keys for blowfish. In: Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 167–180. Springer (2007). https://doi.org/10.1007/978-3-540-74619-5_11

27. Khairallah, M.: Weak keys in the rekeying paradigm: Application to COMET and mixfeed. IACR Trans. Symmetric Cryptol. **2019**(4), 272–289 (2019). https://doi.org/10.13154/TOSC.V2019.I4.272-289
28. Knudsen, L.R.: Iterative characteristics of DES and s^2-des. In: Brickell, E.F. (ed.) CRYPTO '92. vol. 740, pp. 497–511. Springer (1992). https://doi.org/10.1007/3-540-48071-4_35
29. Lai, X., Massey, J.L., Murphy, S.: Markov ciphers and differential cryptanalysis. In: Davies, D.W. (ed.) EUROCRYPT '91. LNCS, vol. 547, pp. 17–38. Springer (1991). https://doi.org/10.1007/3-540-46416-6_2
30. Langford, S.K., Hellman, M.E.: Differential-linear cryptanalysis. In: CRYPTO '94. vol. 839, pp. 17–25. Springer (1994). https://doi.org/10.1007/3-540-48658-5_3
31. Leander, G., Abdelraheem, M.A., Alkhzaimi, H., Zenner, E.: A cryptanalysis of printcipher: The invariant subspace attack. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 206–221. Springer (2011). https://doi.org/10.1007/978-3-642-22792-9_12
32. Leander, G., Minaud, B., Rønjom, S.: A generic approach to invariant subspace attacks: Cryptanalysis of robin, iscream and zorro. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 254–283. Springer (2015). https://doi.org/10.1007/978-3-662-46800-5_11
33. Li, M., Sun, L., Wang, M.: Automated key recovery attacks on round-reduced orthros. In: Batina, L., Daemen, J. (eds.) AFRICACRYPT 2022. LNCS, vol. 13503, pp. 189–213. Springer Nature Switzerland (2022). https://doi.org/10.1007/978-3-031-17433-9_9
34. Liu, F., Isobe, T., Meier, W., Sakamoto, K.: Weak keys in reduced AEGIS and tiaoxin. IACR Trans. Symmetric Cryptol. **2021**(2), 104–139 (2021). https://doi.org/10.46586/TOSC.V2021.I2.104-139
35. Mennink, B.: Weak keys for aez, and the external key padding attack. In: Handschuh, H. (ed.) CT-RSA 2017. LNCS, vol. 10159, pp. 223–237. Springer (2017). https://doi.org/10.1007/978-3-319-52153-4_13
36. Moore, J.H., Simmons, G.J.: Cycle structures of the DES with weak and semi-weak keys. In: Odlyzko, A.M. (ed.) CRYPTO '86. LNCS, vol. 263, pp. 9–32. Springer (1986). https://doi.org/10.1007/3-540-47721-7_2
37. Peyrin, T., Tan, Q.Q.: Mind your path: On (key) dependencies in differential characteristics. IACR Trans. Symmetric Cryptol. **2022**(4), 179–207 (2022). https://doi.org/10.46586/TOSC.V2022.I4.179-207, https://doi.org/10.46586/tosc.v2022.i4.179-207
38. Rohit, R., Sarkar, S.: Diving deep into the weak keys of round reduced ascon. IACR Trans. Symmetric Cryptol. **2021**(4), 74–99 (2021). https://doi.org/10.46586/TOSC.V2021.I4.74-99, https://doi.org/10.46586/tosc.v2021.i4.74-99
39. Sun, L.: A linearisation method for identifying dependencies in differential characteristics: Examining the intersection of deterministic linear relations and nonlinear constraints. Cryptology ePrint Archive, Paper 2024/1849 (2024), https://eprint.iacr.org/2024/1849
40. Sun, L., Wang, W., Wang, M.: More accurate differential properties of LED64 and Midori64. IACR Trans. Symmetric Cryptol. **2018**(3), 93–123 (2018). https://doi.org/10.13154/tosc.v2018.i3.93-123
41. Taka, K., Ishikawa, T., Sakamoto, K., Isobe, T.: An efficient strategy to construct a better differential on multiple-branch-based designs: Application to orthros. In: Rosulek, M. (ed.) CT-RSA 2023. LNCS, vol. 13871, pp. 277–304. Springer (2023). https://doi.org/10.1007/978-3-031-30872-7_11

42. Todo, Y., Leander, G., Sasaki, Y.: Nonlinear invariant attack - practical attack on full scream, iscream, and midori64. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10032, pp. 3–33 (2016). https://doi.org/10.1007/978-3-662-53890-6_1
43. Todo, Y., Leander, G., Sasaki, Y.: Nonlinear invariant attack: Practical attack on full scream, iscream, and midori64. J. Cryptol. **32**(4), 1383–1422 (2019). https://doi.org/10.1007/S00145-018-9285-0

— Supplementary Material —

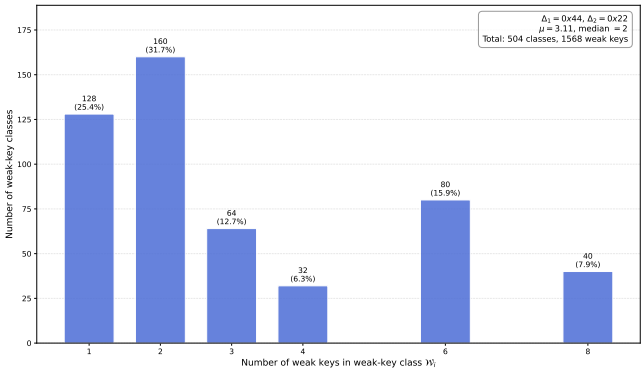


Fig. 8: Weak-key class distribution in Example 1.

A Additional Background

A.1 Probability Distributions

Definition 14 (Normal or Gaussian Distribution). *A random variable X follows a normal distribution with mean μ and variance σ^2 if its probability density function (PDF) is given by:*

$$\phi(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \quad -\infty < x < \infty.$$

We denote this distribution as $X \sim \mathcal{N}(\mu, \sigma^2)$, where $\mathbb{E}[X] = \mu$ and $Var(X) = \sigma^2$.

The CDF of the normal distribution is:

$$\Pr(X \leq x) = \Phi(x) = \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right],$$

where $\operatorname{erf}(\cdot)$ is the error function defined as:

$$\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt.$$

Lemma 3 (Exponential Decay of Normal Tail Probabilities). *For the standard normal CDF $\Phi(z)$ and any $z > 0$, we have:*

$$1 - \Phi(z) < \frac{1}{z\sqrt{2\pi}} e^{-z^2/2}.$$

Consequently, if $P_{FP}(N) = 1 - \Phi(a\sqrt{N} - b)$ for constants $a > 0$ and b, then $P_{FP}(N)$ decreases exponentially in N for sufficiently large N.

Proof. For $z > 0$, the tail probability is $1 - \Phi(z) = \frac{1}{\sqrt{2\pi}} \int_z^\infty e^{-t^2/2} dt$. Since $t > z$ in the integration region, we have $\frac{1}{t} < \frac{1}{z}$. Rewriting the integrand:

$$1 - \Phi(z) = \frac{1}{\sqrt{2\pi}} \int_z^\infty \frac{t}{t} e^{-t^2/2} dt < \frac{1}{z\sqrt{2\pi}} \int_z^\infty t e^{-t^2/2} dt.$$

Computing the integral: $\int_z^\infty t e^{-t^2/2} dt = \left[-e^{-t^2/2} \right]_z^\infty = e^{-z^2/2}$. This establishes the inequality $1 - \Phi(z) < \frac{1}{z\sqrt{2\pi}} e^{-z^2/2}$.

For the second part, substitute $z = a\sqrt{N} - b$. When N is large enough that $a\sqrt{N} > b$:

$$P_{\text{FP}}(N) = 1 - \Phi(a\sqrt{N} - b) < \frac{1}{(a\sqrt{N} - b)\sqrt{2\pi}} e^{-(a\sqrt{N}-b)^2/2} = \mathcal{O}\left(\frac{1}{\sqrt{N}} \right) \cdot e^{-\mathcal{O}(N)}.$$

Definition 15 (Folded Normal Distribution). *Let X be a random variable following a normal distribution with mean μ and variance σ^2, i.e., $X \sim \mathcal{N}(\mu, \sigma^2)$. The folded normal distribution is the distribution of the absolute value of X, i.e., $Y = |X|$, denoted as $Y \sim \mathcal{FN}(\mu, \sigma^2)$.*

The PDF of the folded normal distribution is:

$$f(y) = \frac{1}{\sqrt{2\pi}\sigma} \left[e^{-\frac{(y-\mu)^2}{2\sigma^2}} + e^{-\frac{(y+\mu)^2}{2\sigma^2}} \right], \quad y \geq 0.$$

The CDF of the folded normal distribution is:

$$\Pr(Y \leq y) = F(y) = \Phi\left(\frac{y - \mu}{\sigma} \right) - \Phi\left(\frac{-y - \mu}{\sigma} \right),$$

where $\Phi(\cdot)$ is the CDF of the standard normal distribution.

The expected value and variance of a folded normal distribution are:

$$\mathbb{E}[Y] = \sigma \cdot 2 \cdot \phi\left(\frac{\mu}{\sigma} \right) + \mu \left[1 - 2\Phi\left(-\frac{\mu}{\sigma} \right) \right],$$

$$Var[Y] = \mu^2 + \sigma^2 - \mathbb{E}[Y]^2,$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ are the PDF and CDF of the standard normal distribution, respectively.

Definition 16 (Poisson-Binomial Distribution). *Let $X_1, \ldots, X_n$ be independent Bernoulli random variables with $X_i \sim Bernoulli(p_i)$ for $i = 1, \ldots, n$, where $p_i \in [0, 1]$ are potentially different success probabilities. The sum $Y = \sum_{i=1}^n X_i$ follows a **Poisson-Binomial distribution** with parameters $(p_1, \ldots, p_n)$.*

The PMF is:

$$\Pr(Y = k) = \sum_{A \in \mathcal{S}_k} \prod_{i \in A} p_i \prod_{j \in A^c} (1 - p_j),$$

where $\mathcal{S}_k$ is the set of all subsets of k integers from $\{1, 2, \ldots, n\}$.

The mean and variance are:

$$\mathbb{E}[Y] = \sum_{i=1}^{n} p_i, \quad Var(Y) = \sum_{i=1}^{n} p_i(1-p_i).$$

When all $p_i = p$ are equal, this reduces to the Binomial(n, p) distribution. When $p_i \ll 1$ for all i and $\sum_{i=1}^{n} p_i = \lambda$ is moderate, the Poisson-Binomial distribution can be approximated by a Poisson(λ) distribution.

A.2 Orthogonality Relations of Walsh Basis Functions

Lemma 4 (Orthogonality Relations of Walsh Basis Functions). *Let $\chi_\omega(x) = (-1)^{\omega \cdot x}$ denote the Walsh character functions over $\mathbb{F}_2^n$. Then, the following orthogonality relations hold:*

$$\sum_{\omega \in \mathbb{F}_2^n} \chi_\omega(x) = \begin{cases} 2^n & \text{if } x = 0, \\ 0 & \text{otherwise,} \end{cases} \qquad \sum_{x \in \mathbb{F}_2^n} \chi_\omega(x)\chi_{\omega'}(x) = \begin{cases} 2^n & \text{if } \omega = \omega', \\ 0 & \text{otherwise.} \end{cases}$$

That is, $\sum_{\omega \in \mathbb{F}_2^n} (-1)^{\omega \cdot x} = 2^n \delta_0(x)$ and $\sum_{x \in \mathbb{F}_2^n} (-1)^{(\omega+\omega') \cdot x} = 2^n \delta_0(\omega+\omega')$, where $\delta_x(y)$ is the Kronecker delta function.

A.3 Proof of Lemma 1

Here, we prove Lemma 1

Proof. We start from the right hand side of Equation 4:

$$\begin{aligned} 2^{-m} \sum_{a \in \mathbb{F}_2^m} (-1)^{a \cdot y} \mathrm{Cr}(a \cdot F) &= 2^{-m} \sum_{a \in \mathbb{F}_2^m} (-1)^{a \cdot y} 2^{-n} \sum_{x \in \mathbb{F}_2^n} (-1)^{a \cdot F(x)} \\ &= 2^{-m} 2^{-n} \sum_{x \in \mathbb{F}_2^n} \sum_{a \in \mathbb{F}_2^m} (-1)^{a \cdot (y + F(x))} \end{aligned}$$

According to Lemma 4, we know that $\sum_{a \in \mathbb{F}_2^m} (-1)^{a \cdot (y+F(x))} = 2^m$ if $y = F(x)$, and 0 otherwise. As a result, we have:

$$2^{-m} \sum_{a \in \mathbb{F}_2^m} (-1)^{a \cdot y} \mathrm{Cr}(a \cdot F) = 2^{-n} \sum_{x \in \mathbb{F}_2^n} \delta_y(F(x)) = \Pr(F(X) = y).$$

For the second part, we replace the probability in the right-hand side of Equation 5 with the first part.

$$\begin{aligned} \sum_{y \in \mathbb{F}_2^m} (-1)^{a \cdot y} \Pr(F(X) = y) &= 2^{-m} \sum_{b \in \mathbb{F}_2^m} \sum_{y \in \mathbb{F}_2^m} (-1)^{(a+b) \cdot y} \mathrm{Cr}(b \cdot F) \\ &= 2^{-m} \sum_{b \in \mathbb{F}_2^m} \delta_a(b) \cdot 2^m \cdot \mathrm{Cr}(b \cdot F) = \mathrm{Cr}(a \cdot F). \end{aligned}$$

A.4 LLR Histogram Identity

Lemma 5 (LLR histogram identity). *For samples $x_1, \ldots, x_N \in \Omega$, counts $c(x) = \sum_{i=1}^{N} \delta_x(x_i)$, and empirical distribution $\widehat{p}_N(x) = c(x)/N$, we have*

$$\sum_{i=1}^{N} \log \tfrac{p(x_i)}{q(x_i)} \;=\; \sum_{x\in\Omega} c(x) \log \tfrac{p(x)}{q(x)} \;=\; N \sum_{x\in\Omega} \widehat{p}_N(x) \log \tfrac{p(x)}{q(x)}. \tag{23}$$

Proof. The identity follows by expanding each term using the Kronecker delta and exchanging the order of summation. Let $c(x) = \sum_{i=1}^{N} \delta_x(x_i)$ denote the counts and $\widehat{p}_N(x) = c(x)/N$ the empirical distribution. Then

$$\sum_{i=1}^{N} \log \frac{p(x_i)}{q(x_i)} = \sum_{i=1}^{N} \sum_{x\in\Omega} \delta_x(x_i) \log \frac{p(x)}{q(x)} = \sum_{x\in\Omega} c(x) \log \frac{p(x)}{q(x)} = N \sum_{x\in\Omega} \widehat{p}_N(x) \log \frac{p(x)}{q(x)},$$

establishing the claim.

A.5 Proof of Theorem 1

Here, we prove Theorem 1.

Proof. We start with the left-hand side of the relation for capacity. Expanding the square, we get:

$$\begin{aligned}\mathrm{Cp}(F(X)) =& 2^m \sum_{y\in\mathbb{F}_2^m} \left(\Pr(F(X)=y)^2 - 2\cdot\Pr(F(X)=y)\cdot 2^{-m} + 2^{-2m}\right) \\ =& 2^m \left(\sum_{y\in\mathbb{F}_2^m} \Pr(F(X)=y)^2 - 2\cdot 2^{-m} \sum_{y\in\mathbb{F}_2^m} \Pr(F(X)=y) + 2^{-m} \right).\end{aligned}$$

The second sum simplifies since $\sum_{y\in\mathbb{F}_2^m} \Pr(F(X)=y) = 1$, so:

$$\mathrm{Cp}(F(X)) = 2^m \sum_{y\in\mathbb{F}_2^m} \Pr(F(X)=y)^2 - 1.$$

Now, we substitute $\Pr(F(X)=y)$ from Theorem 1, and we get:

$$\begin{aligned}\Pr(F(X)=y)^2 =& \left(2^{-m} \sum_{a\in\mathbb{F}_2^m} (-1)^{a\cdot y} \mathrm{Cr}(a\cdot F) \right)^2 \\ =& 2^{-2m} \sum_{a,a'\in\mathbb{F}_2^m} (-1)^{(a+a')\cdot y} \mathrm{Cr}(a\cdot F)\mathrm{Cr}(a'\cdot F).\end{aligned}$$

Now summing over $y \in \mathbb{F}_2^m$, we use the orthogonality of the characters:

$$\sum_{y\in\mathbb{F}_2^m} (-1)^{(a+a')\cdot y} = \begin{cases} 2^m & \text{if } a = a', \\ 0 & \text{otherwise.} \end{cases}$$

Thus, the sum simplifies to:

$$\sum_{y \in \mathbb{F}_2^m} \Pr(F(X) = y)^2 = 2^{-m} \sum_{a \in \mathbb{F}_2^m} \mathrm{Cr}^2(a \cdot F).$$

Finally, substituting this back into the expression for $\mathrm{Cp}(F(X))$, and considering that $\mathrm{Cr}(0 \cdot F) = 1$, we have

$$\mathrm{Cp}(F(X)) = \sum_{a \in \mathbb{F}_2^m} \mathrm{Cr}^2(a \cdot F) - 1 = \sum_{a \in \mathbb{F}_2^m, a \neq 0} \mathrm{Cr}^2(a \cdot F).$$

A.6 KL Divergence Bounds and Approximations

Lemma 6 (Upper Bound on KL Divergence). *Let $p(x)$ and $q(x)$ be two probability mass functions. Then,*

$$D_{KL}(p \parallel q) \leq \mathrm{Cp}(p, q).$$

Proof. Since for any $x > 0$ we have $\log(x) < x - 1$, we get

$$D_{\mathrm{KL}}(p \parallel q) = \sum_x p(x) \log \frac{p(x)}{q(x)} \leq \sum_x p(x) \left(\frac{p(x)}{q(x)} - 1 \right) = \mathrm{Cp}(p, q).$$

Lemma 7 (Approximate Kullback-Leibler Divergence [23,3]). *Let $p(x)$ and $q(x)$ be two probability mass functions with $|p(x) - q(x)| \ll q(x)$ for all x. Then,*

$$D_{KL}(p \parallel q) \approx \tfrac{1}{2}\mathrm{Cp}(p, q) + \mathcal{O}(\max_x |p(x) - q(x)|^3).$$

A.7 Brief Specification of Orthros

Orthros [4] is a 128-bit low-latency keyed pseudo-random function designed by Banik et al. and presented at FSE 2022. Its structure consists of the sum of two 128-bit keyed permutations, referred to as Branch1 and Branch2, as in Figure 9.

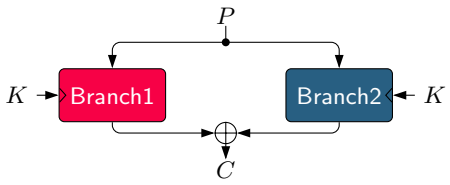


Fig. 9: The Orthros PRF construction.

Both branches use the same 128-bit key K. A 128-bit plaintext P is initially copied into two 128-bit states, which are then encrypted separately under

Branch1 and Branch2 using K. Finally, a 128-bit ciphertext C is produced by XORing the two 128-bit outputs from each branch.

Branch1 and Branch2 are both 12-round SPN-based ciphers. Each round function includes an S-box layer, a bit or nibble permutation, matrix multiplication, round key addition, and constant addition. Additionally, a key addition operation, where whitening keys are XORed with the state, occurs before the first round in each branch.

The two branches share the same S-box layer, composed of 32 parallel 4-bit S-boxes $\mathcal{S}$, as shown in Figure 10a. However, the bit and nibble permutations differ between the branches. Branch1 uses the bit permutation P_{br1} and the nibble permutation P_{nr1}, while Branch2 uses P_{br2} and P_{nr2}, respectively. For example, in Branch1, the most significant bit of the state is denoted as the 0-th bit. The i-th bit of the state before P_{br1} will be the $P_{br1}(i)$-th bit of the state after P_{br1}. Similarly, the j-th nibble of the state before P_{nr1} is placed in the $P_{nr1}(j)$-th nibble after P_{nr1}. The permutations are specified in Figure 11.

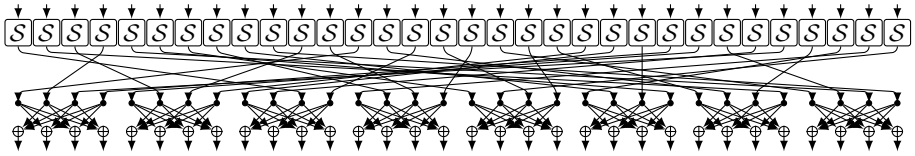
Note that the bit permutation is used in the first four rounds, and the nibble permutation is adopted in the next seven rounds, while no permutation is utilized in the final round. The matrix multiplication is also the same for both branches, where eight 4×4 matrices ($\mathbf{M_b}$) are applied on these 32 nibbles. The matrix $\mathbf{M_b}$ is an almost MDS matrix as defined below. In the final round, there is no matrix multiplication.

$$\mathbf{M_b} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

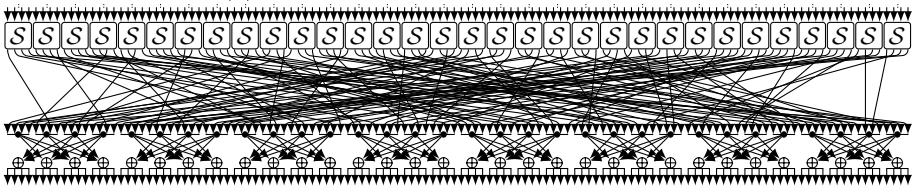
Orthros utilizes two linear key schedules, KSF1 and KSF2, which consist solely of bit permutations P_{bk1} and P_{bk2}. In KSF1, P_{bk1} is employed to generate the whitening and round keys used in Branch1, while in KSF2, P_{bk2} is used for Branch2. Given the 128-bit key K, KSF1 (respectively KSF2) first updates its value using P_{bk1} (respectively P_{bk2}), and then generates the whitening key WK1 (respectively WK2). Subsequently, KSF1 (respectively KSF2) outputs each round key $RK1_r$ (respectively $RK2_r$) after updating its state using the permutation. Descriptions of P_{bk1} and P_{bk2} are provided in Figure 12.

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$\mathcal{S}(x)$	1	0	2	4	3	8	6	d	9	a	b	e	f	c	7	5

(a) 4-bit S-box used in Orthros

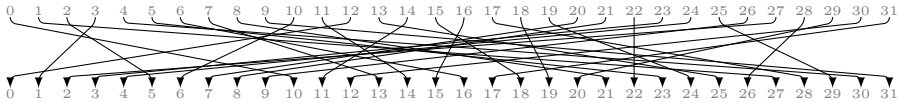


(b) Rounds 1 to 4 with nibble permutation.



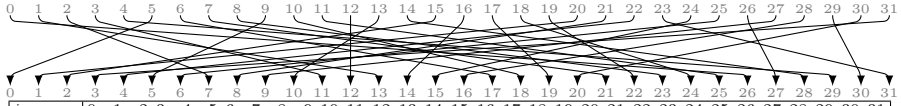
(c) Rounds 5 to 11 with bit permutation.

Fig. 10: Round functions of Branch1 (without round key addition); round 12 omits the linear layer. Branch2 uses different nibble and bit permutations.



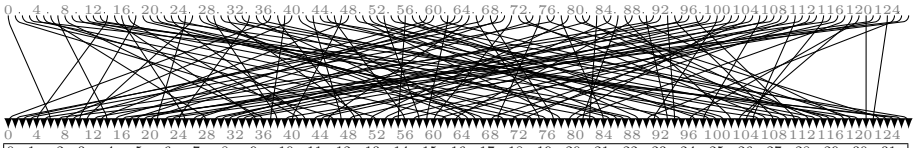
i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$P_{nr1}(i)$	10	27	5	1	30	23	16	13	21	31	6	14	0	25	11	18	15	28	19	24	7	8	22	3	4	29	9	2	26	20	12	17

(a) Nibble permutation P_{nr1} for rounds 1 to 4 of Branch1.



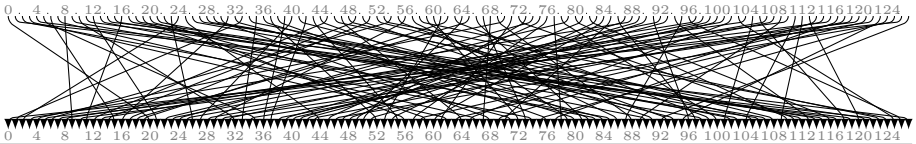
i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$P_{nr2}(i)$	26	13	7	11	29	0	17	21	23	5	18	25	12	10	28	2	14	19	24	22	1	8	4	31	15	6	27	9	16	30	20	3

(b) Nibble permutation P_{nr2} for rounds 1 to 4 of Branch2.



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

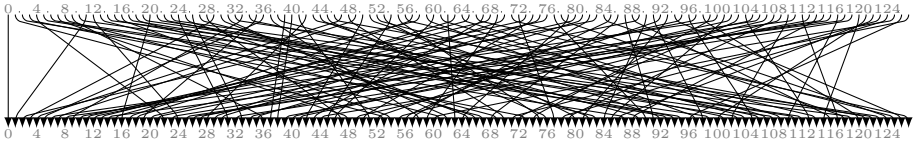
(c) Bit permutation P_{br1} for rounds 5 to 11 of Branch1.



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

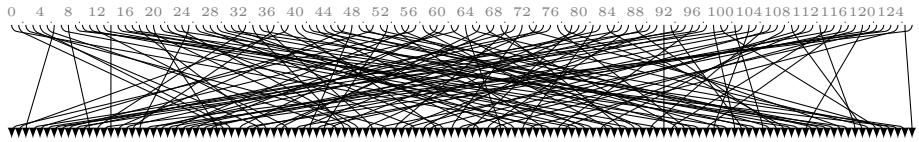
(d) Bit permutation P_{br2} for rounds 5 to 11 of Branch2.

Fig. 11: Permutation layers of the Orthros round functions.



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

(a) Bit permutation P_{bk1} for the key schedule of Branch1.



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

(b) Bit permutation P_{bk2} for the key schedule of Branch2.

Fig. 12: Permutation layers of the Orthros key schedule.

B DLCT and QDLCT of Orthros S-box

Table 6: DLCT of the Orthros S-box.

$\delta \setminus \omega$	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
1	16	-8	-8	0	8	-8	-8	8	8	0	0	-8	0	0	0	0
2	16	0	8	0	0	0	-8	0	8	0	0	0	-8	0	-16	0
3	16	0	-8	-8	0	0	0	0	0	-8	8	0	0	8	0	-8
4	16	8	0	0	0	-8	0	0	0	8	-8	-8	-8	0	0	0
5	16	-8	0	-8	0	0	0	0	0	0	0	0	0	8	-16	8
6	16	0	0	0	-8	0	0	-8	0	0	0	0	8	0	0	-8
7	16	0	0	0	-8	0	0	-8	0	-8	-8	0	0	0	16	0
8	16	8	8	0	8	0	0	-8	-8	-8	0	0	-8	-8	0	0
9	16	-8	-8	8	0	0	0	-8	-8	0	0	0	0	0	0	8
a	16	0	0	-8	0	-8	0	0	-8	0	8	8	0	0	0	-8
b	16	0	-8	0	0	8	0	0	-8	8	0	-8	0	-8	0	0
c	16	0	0	-8	0	0	-8	0	0	-8	0	0	8	0	0	0
d	16	-8	0	0	0	0	8	0	0	0	-8	0	0	-8	0	0
e	16	0	0	0	-8	8	8	8	0	0	-8	-8	-8	-8	0	0
f	16	0	0	8	-8	-8	-8	0	0	0	0	8	0	0	0	-8

Table 7: QDLCT of the Orthros S-box.

$\delta \setminus \omega$	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	65536	65536	65536	65536	65536	65536	65536	65536	65536	65536	65536	65536	65536	65536	65536	65536
1	65536	-3904	-3776	-3712	-4608	-4032	-3712	-5312	-4864	-3520	-4736	-4416	-5696	-3584	-6400	-3264
2	65536	-4032	-4288	-3968	-4672	-4608	-4864	-4544	-5504	-3776	-3648	-3456	-5184	-3456	-6016	-3520
3	65536	-4416	-4288	-3712	-4416	-4992	-4672	-4352	-4288	-4608	-4800	-4096	-4928	-4096	-4096	-3776
4	65536	-3712	-4608	-4992	-3456	-4352	-4032	-4160	-4672	-3776	-4352	-5376	-4992	-4224	-4992	-3840
5	65536	-4480	-4352	-4096	-5824	-3904	-4032	-4800	-5056	-4544	-4800	-3648	-4352	-3840	-4096	-3712
6	65536	-4928	-4032	-4608	-4096	-4672	-5248	-3776	-3328	-4800	-4224	-4288	-4416	-3968	-5120	-4032
7	65536	-6144	-5248	-5376	-4416	-3520	-4032	-3264	-2624	-4928	-4160	-5056	-3584	-3712	-4864	-4608
8	65536	-4096	-4736	-3840	-3968	-4096	-4160	-3904	-5440	-3136	-3328	-4352	-4736	-5504	-4992	-5248
9	65536	-4608	-3840	-4224	-5056	-3776	-3520	-4928	-4032	-4672	-5056	-4288	-3328	-5504	-3328	-5376
a	65536	-3392	-4032	-3840	-4352	-5568	-4736	-3648	-4096	-4416	-5504	-4416	-4928	-5504	-2048	-5056
b	65536	-3328	-3968	-3968	-4032	-4288	-4544	-6080	-5312	-4544	-4032	-4288	-3456	-5248	-3328	-5120
c	65536	-4864	-4352	-3584	-4864	-3968	-4480	-5120	-4736	-4352	-3840	-3712	-4864	-3328	-5888	-3584
d	65536	-4800	-4672	-5376	-4480	-3904	-4224	-3648	-3712	-4544	-4352	-4800	-3392	-4864	-3584	-5184
e	65536	-4544	-5312	-5632	-3648	-4224	-4480	-4416	-4480	-5056	-3648	-4864	-3008	-4224	-3200	-4800
f	65536	-4288	-4032	-4608	-3648	-5632	-4800	-3584	-3392	-4864	-5056	-4480	-4672	-4480	-3584	-4416

C Modeling the S-box Using the S-box Analyzer

In this section, we describe how to model the propagation of determinsitic differential and linear trails through a given S-box or vectorial Boolean function. We model the propagation of deterministic trails through a given building block as a Boolean function, and use logic minimization algorithms [12] to derive the corresponding near-optimized MILP or SAT constraints. We illustrate the idea with a basic example. Assume that we need a small set of MILP or SAT constraints to detect the overlap between deterministic forward and backward difference propagations. In other words, let x denote the deterministic forward difference and y the deterministic backward linear mask at a certain bit position. We aim to model function $F : \{-1, 0, 1\}^2 \to \{0, 1\}$:

$$F(x, y) = \begin{cases} 1 & \text{if } (x, y) \in \{(-1, -1), (-1, 1), (1, -1)\}, \\ 0 & \text{otherwise.} \end{cases}$$

Let us encode x, y, and $F(x, y)$ by (a_0, a_1), (b_0, b_1), and c_0, respectively, where a_0, a_1, b_0, b_1, and c_0 are binary variables. We then express the function F as a set of 9 integer vectors $(x, y, F(x, y))$, or equivalently as 9 binary vectors $(a_0, a_1, b_0, b_1, c_0) \in \mathbb{F}_2^5$, as shown in Figure 13. Once the function is represented as a set of binary vectors, we can encode its characteristic function as a Boolean function, which can then be translated into CNF, MILP, or CP constraints. To simplify this process, we have added a feature to the `S-box Analyzer` tool that takes a set of binary vectors as input and automatically generates the corresponding CNF, MILP, or CP constraints (see Listing 1.1).

x	y	$F(x,y)$	a_0	a_1	b_0	b_1	c_0
-1	-1	1	1	0	1	0	1
-1	0	0	1	0	0	0	0
-1	1	1	1	0	0	1	1
0	-1	0	0	0	1	0	0
0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0
1	-1	1	0	1	1	0	1
1	0	0	0	1	0	0	0
1	1	0	0	1	0	1	0

Fig. 13: Valid points of the overlap function

```
from sboxanalyzer import SboxAnalyzer as sa
S=[(1, 0, 1, 0, 1), (1, 0, 0, 0, 0),
(1, 0, 0, 1, 1), (0, 0, 1, 0, 0),
(0, 0, 0, 0, 0), (0, 0, 0, 1, 0),
(0, 1, 1, 0, 1), (0, 1, 0, 0, 0),
(0, 1, 0, 1, 0)]
cnf,milp,cp=sa.encode_set_of_binary_vectors
(S, input_variables=['a0','a1','b0','b1','c0'])
milp
['- a1 - b1 - c0 >= -2',
 'b0 + b1 - c0 >= 0',
 'a0 + a1 - c0 >= 0',
 '- a0 - b1 + c0 >= -1',
 '- a1 - b0 + c0 >= -1',
 '- a0 - b0 + c0 >= -1',
 '- b0 - b1 >= -1',
 '- a0 - a1 >= -1']
```

Listing 1.1: modeling set of binary vectors

Listing 1.2 and Listing 1.3 illustrate the encoding of differential and linear behaviors of the S-box, in S-box Analyzer, respectively.

```
# Import the S-box Analyzer and define the S-box
from sboxanalyzer import *
sa = SboxAnalyzer([0x1,0x0,0x2,0x4,0x3,0x8,0x6,0xd,0x9,0xa,0xb,0xe,0xf,0xc,0x7,0x5])
# Model differential propagation with probabilities
cnf, milp, cp = sa.minimized_diff_constraints()
Number of constraints: 64
Input: a0||a1||a2||a3; msb: a0
Output: b0||b1||b2||b3; msb: b0
Weight: 3.0000 p0 + 2.0000 p1

# Model forward differential propagation without probabilities
cnf, milp,cp = sa.minimized_diff_constraints(subtable='star')
Number of constraints: 54
Input: a0||a1||a2||a3; msb: a0
Output: b0||b1||b2||b3; msb: b0

# Model deterministic differential propagation by CP constraints (forward)
temp = sa.encode_deterministic_differential_behavior()
cp = sa.generate_cp_constraints(temp); print(cp)
Input: a0||a1||a2||a3; msb: a0
Output: b0||b1||b2||b3; msb: b0
if (a0 == 0 /\ a1 == 0 /\ a2 == 0 /\ a3 == 0) then (b0 = 0 /\ b1 = 0 /\ b2 = 0 /\ b3 = 0)
else (b0 = -1 /\ b1 = -1 /\ b2 = -1 /\ b3 = -1)
endif

# Model deterministic differential propagation by MILP constraints (forward)
cnf, milp, cp = sa.minimized_deterministic_diff_constraints(); pretty_print(milp)
Number of constraints: 20
Input: a0||a1||a2||a3||a4||a5||a6||a7; msb: a0
Output: b0||b1||b2||b3||b4||b5||b6||b7; msb: b0
- a0 + b6 >= 0
- a1 + b6 >= 0
- a2 + b6 >= 0
- a3 + b6 >= 0
- a4 + b6 >= 0
- a5 + b6 >= 0
- a6 + b6 >= 0
- a7 + b6 >= 0
- b0 + b1 + b2 + b3 + b5 + b7 >= 0
b0 + b1 + b3 - b4 + b5 + b7 >= 0
b1 + b3 + b4 + b5 - b6 + b7 >= 0
a0 + a1 + a2 + a3 + a4 + a5 + a6 + a7 + b1 - b2 + b3 + b5 + b7 >= 0
- a6 - a7 >= -1
- a4 - a5 >= -1
- a2 - a3 >= -1
- a0 - a1 >= -1
- b7 >= 0
- b5 >= 0
- b3 >= 0
- b1 >= 0
```

Listing 1.2: Encoding differential behavior of S-box

```
# Import the S-box Analyzer and define the S-box
from sboxanalyzer import *
sa = SboxAnalyzer([0x1,0x0,0x2,0x4,0x3,0x8,0x6,0xd,0x9,0xa,0xb,0xe,0xf,0xc,0x7,0x5])
sai = SboxAnalyzer(sa.inverse())
# Model forward linear mask propagation with correlations
cnf, milp, cp = sa.minimized_linear_constraints()
Number of constraints: 71
Input: a0||a1||a2||a3; msb: a0
Output: b0||b1||b2||b3; msb: b0
Weight: 4.0000 p0 + 2.0000 p1

# Model backward linear mask propagation withoout correlations
cnf, milp,cp = sai.minimized_linear_constraints(subtable='star')
Number of constraints: 44
Input: a0||a1||a2||a3; msb: a0
Output: b0||b1||b2||b3; msb: b0

# Model deterministic linear mask propagation by CP constraints (backward)
temp = sai.encode_deterministic_linear_behavior()
cp = sai.generate_cp_constraints(temp); print(cp)
Input: a0||a1||a2||a3; msb: a0
Output: b0||b1||b2||b3; msb: b0
if (a0==0/\a1==0/\a2==0/\a3==0) then (b0=0/\b1=0/\b2=0/\b3=0)
elseif (a0==1/\a1==1/\a2==1/\a3==0) then (b0=-1/\b1=-1/\b2=1/\b3=-1)
else (b0=-1/\b1=-1/\b2=-1/\b3=-1)
endif

# Model deterministic linear mask propagation by MILP constraints (backward)
cnf, milp, cp = sai.minimized_deterministic_lin_constraints(); pretty_print(milp)
Number of constraints: 23
Input: a0||a1||a2||a3||a4||a5||a6||a7; msb: a0
Output: b0||b1||b2||b3||b4||b5||b6||b7; msb: b0
- a6 + b4 >= 0
- a7 + b4 >= 0
a5 - b5 >= 0
- b4 - b5 >= -1
- a0 + b6 >= 0
- a2 + b6 >= 0
- a4 + b6 >= 0
- a5 + b6 >= 0
- b4 + b6 >= 0
- a1 + a3 + b4 >= 0
a1 - b2 + b4 >= 0
- a3 + b4 + b5 >= 0
- b0 + b1 + b2 + b3 + b7 >= 0
b0 + b1 + b3 - b6 + b7 >= 0
- a1 - a3 - a5 + a6 + a7 + b5 >= -2
a0 + a1 + a2 + a3 + a4 + a5 + a6 + a7 - b2 >= 0
- a6 - a7 >= -1
- a4 - a5 >= -1
- a2 - a3 >= -1
- a0 - a1 >= -1
- b7 >= 0
- b3 >= 0
- b1 >= 0
```

Listing 1.3: Encoding linear behavior of S-box

D Alternative Classification Method Using Maximum Statistic

In Section 4, we presented a classification method based on the LLR statistic to determine whether a secret key belongs to a weak-key class. Here we describe an alternative approach based on the maximum statistic, which offers complementary theoretical properties and is particularly useful when strict FWER control is required or when only a few assignments carry strong correlations.

A key difference between the two methods is the distinguisher structure: while the LLR method exploits a multidimensional DL distinguisher using all nonzero output masks simultaneously (e.g., all 15 masks for a 4-bit output space), the maximum statistic approach uses a single DL distinguisher with one output mask ω. This design choice simplifies the statistical analysis and enables rigorous FWER control, but typically requires more data to achieve comparable detection power.

D.1 Statistical Framework for Maximum Statistic

Normalization and Distribution Under Null Hypothesis Recall from Section 4 that for a given weak-key class $\mathcal{W}_i$ with good input assignment set $\mathcal{X}_i$, we collect N samples per assignment. Let $\widehat{\boldsymbol{c}}_x$ denote the empirical autocorrelation for assignment $x \in \mathcal{X}_i$, estimated from N samples.

Definition 17 (Normalized Correlation). *For each tested assignment x with empirical autocorrelation $\widehat{\boldsymbol{c}}_x$ computed from N samples, define the normalized statistic*

$$z_x = \frac{\widehat{\boldsymbol{c}}_x}{\sqrt{1/N}} = \widehat{\boldsymbol{c}}_x \sqrt{N}.$$

Under the null hypothesis H_0 (key is strong), the empirical autocorrelation $\widehat{\boldsymbol{c}}_x$ behaves as if sampled from an ideal cipher. By Equation 10, we have $\widehat{\boldsymbol{c}}_x \sim \mathcal{N}(0, 1/N)$, which implies $z_x \sim \mathcal{N}(0,1)$. Since we observe the absolute value $|\widehat{\boldsymbol{c}}_x|$, the relevant distribution is the folded normal (equivalently, half-normal when the mean is zero).

Lemma 8 (Half-Normal Distribution Under H_0). *Under the null hypothesis H_0 (strong-key), the normalized absolute correlations $|z_x|$ for $x \in \mathcal{X}_i$ are i.i.d. half-normal random variables with density*

$$f(z) = \sqrt{\frac{2}{\pi}} \exp\left(-\frac{z^2}{2}\right), \quad z \geq 0.$$

The CDF is $F(z) = erf(z/\sqrt{2})$, where erf is the error function.

Under the alternative hypothesis H_1 (key belongs to weak-key class $\mathcal{W}_i$), assignments in $\mathcal{X}_i$ exhibit nonzero expected autocorrelation. Let $\boldsymbol{c}$ denote the analytically derived autocorrelation for the distinguisher (which may be positive

or negative; we work with its absolute value). The empirical correlation $\widehat{\boldsymbol{c}}_x$ satisfies $\widehat{\boldsymbol{c}}_x \sim \mathcal{N}(\pm|\boldsymbol{c}|, 1/N)$ depending on the correlation sign. The standardized statistic is $z_x = \widehat{\boldsymbol{c}}_x\sqrt{N} \sim \mathcal{N}(\pm|\boldsymbol{c}|\sqrt{N}, 1)$. Since we observe $|z_x|$, for detection purposes we use $|z_x| \sim \mathcal{FN}(|\boldsymbol{c}|\sqrt{N}, 1)$, which is the same regardless of correlation sign.

Remark 3 (Independence Assumption). The i.i.d. assumption is approximate because, as described in Section 4, all assignments in $\mathcal{X}_i$ share the same non-active nibble patterns. For large N and many non-active nibbles, the dependence is weak and the half-normal approximation remains useful for threshold selection. For a fully rigorous treatment without independence assumptions, one could use the union bound $\Pr(M_s > q) \leq s \cdot 2[1 - \Phi(q)]$ at the cost of a more conservative threshold.

Maximum Statistic Rather than using the LLR statistic, the maximum statistic approach bases the decision on the largest observed normalized correlation.

Definition 18 (Maximum Statistic). *Let $s = |\mathcal{X}_i|$ denote the number of tested assignments for weak-key class i. The maximum statistic is*

$$M_s = \max_{x \in \mathcal{X}_i} |z_x|.$$

The distribution of M_s under H_0 can be derived from the order statistics of half-normal samples.

Lemma 9 (Distribution of Maximum Statistic Under H_0). *Under the null hypothesis H_0, the maximum statistic M_s has CDF*

$$F_{M_s}(q) = \Pr(M_s \leq q \mid H_0) = \left[erf\left(\frac{q}{\sqrt{2}}\right)\right]^s,$$

and tail probability

$$\Pr(M_s > q \mid H_0) = 1 - \left[erf\left(\frac{q}{\sqrt{2}}\right)\right]^s.$$

For large q with $s \cdot [1 - \Phi(q)] \ll 1$, the union bound approximation gives

$$\Pr(M_s > q \mid H_0) \approx 2s[1 - \Phi(q)],$$

where Φ is the standard normal CDF.

Proof. Since $|z_x|$ are i.i.d. half-normal (under the approximation discussed in the remark above), the maximum $M_s = \max_x |z_x|$ satisfies

$$\Pr(M_s \leq q) = \prod_{x=1}^{s} \Pr(|z_x| \leq q) = [F(q)]^s = \left[\operatorname{erf}\left(\frac{q}{\sqrt{2}}\right)\right]^s.$$

The tail probability follows by complementarity. For the union bound approximation, note that $\Pr(|z_x| > q) = 2[1 - \Phi(q)]$ for a half-normal variable, so when $s \cdot \Pr(|z_x| > q) \ll 1$, we have $\Pr(M_s > q) \approx s \cdot \Pr(|z_x| > q) = 2s[1 - \Phi(q)]$.

D.2 Threshold Determination and Error Control

Family-Wise Error Rate Control When testing multiple weak-key classes, we must control the FWER (see Definition 12). A simple approach is the Bonferroni correction: test each hypothesis at level α/ℓ, where ℓ is the total number of weak-key classes. However, when using the maximum statistic, we have a more refined approach.

Theorem 5 (Maximum Statistic Threshold). *Let $s_i = |\mathcal{X}_i|$ denote the number of tested assignments for weak-key class $\mathcal{W}_i$. To control the Type I error probability for class i at level α_i, i.e., $\Pr(M_{s_i} > q_i \mid H_0) \leq \alpha_i$, set*

$$q_i = \Phi^{-1}\left(1 - \frac{\alpha_i}{2s_i}\right),$$

where Φ^{-1} is the inverse CDF of the standard normal distribution, and the factor of 2 accounts for the half-normal distribution. For FWER control across ℓ tests at level α, use the Bonferroni correction: set $\alpha_i = \alpha/\ell$ for each test.

Proof. Under H_0, each $|z_x|$ follows a half-normal distribution. For a standard normal random variable $Z \sim \mathcal{N}(0,1)$, we have $\Pr(|Z| > q) = 2[1 - \Phi(q)]$. For the maximum of s_i such variables, using the union bound approximation:

$$\Pr(M_{s_i} > q \mid H_0) \approx 2s_i[1 - \Phi(q)].$$

Setting this equal to α_i and solving for q_i yields the stated threshold. For FWER control, the Bonferroni correction ensures $\sum_{i=1}^{\ell} \Pr(M_{s_i} > q_i \mid H_0) \leq \sum_{i=1}^{\ell} \alpha_i = \alpha$.

Remark 4. In the visualization code used for exploratory analysis, we apply $\alpha = 0.01$ with a single global threshold computed using s equal to the total number of input assignments tested in each individual experiment. This approach treats all assignments uniformly and provides conservative control against spurious detections under the strong-key null.

Type I and Type II Errors

Definition 19 (Error Rates for Maximum Statistic Test). *For weak-key class $\mathcal{W}_i$ with threshold q_i:*

- ***Type I Error (False Positive):** The probability of incorrectly classifying a strong-key as belonging to weak-key class $\mathcal{W}_i$:*

$$\alpha_i = \Pr(M_{s_i} > q_i \mid H_0) \approx 2s_i[1 - \Phi(q_i)].$$

- ***Type II Error (False Negative):** The probability of failing to detect a key that truly belongs to weak-key class $\mathcal{W}_i$:*

$$\beta_i = \Pr(M_{s_i} \leq q_i \mid H_1).$$

– ***Detection Probability (Power):*** *The probability of correctly detecting a weak-key:*

$$P^i_{\text{detect}} = 1 - \beta_i = \Pr(M_{s_i} > q_i \mid H_1).$$

Under H_1, the distribution of M_{s_i} is more complex because different assignments may have different signal strengths. For simplicity, assume all assignments in $\mathcal{X}_i$ have the same expected correlation magnitude $|\boldsymbol{c}|$. Then each $|z_x| \sim \mathcal{FN}(|\boldsymbol{c}|\sqrt{N}, 1)$. When $|\boldsymbol{c}|\sqrt{N} > 0$ is sufficiently large, the folded normal distribution concentrates on the positive side, and we can approximate $\Pr(|z_x| \leq q_i) \approx \Phi(q_i - |\boldsymbol{c}|\sqrt{N})$, where Φ is the standard normal CDF. Under the independence assumption[2], the detection probability can be expressed as

$$P^i_{\text{detect}} = \Pr(M_{s_i} > q_i \mid H_1) = 1 - \Pr(M_{s_i} \leq q_i \mid H_1) = 1 - \prod_{x \in \mathcal{X}_i} \Pr(|z_x| \leq q_i \mid H_1),$$

which simplifies to

$$P^i_{\text{detect}} \approx 1 - \left[\Phi\left(q_i - |\boldsymbol{c}|\sqrt{N}\right)\right]^{s_i}.$$

For large s_i and strong signal $|\boldsymbol{c}|\sqrt{N} \gg q_i$, at least one assignment will exceed the threshold with high probability.

Asymptotic behavior as N grows. As the number of samples per assignment N increases, the normalized correlations concentrate around their means. Under H_0 (strong-key), each $|z_x|$ follows a half-normal distribution regardless of N, so the Type I error probability $\alpha_i = \Pr(M_{s_i} > q_i \mid H_0)$ remains fixed for a given threshold q_i. Under H_1 (weak-key), each $|z_x|$ has location parameter $|\boldsymbol{c}|\sqrt{N}$ growing with N. For fixed threshold q_i, the detection probability $P^i_{\text{detect}} = 1 - [\Phi(q_i - |\boldsymbol{c}|\sqrt{N})]^{s_i}$ approaches 1 as $N \to \infty$ because $q_i - |\boldsymbol{c}|\sqrt{N} \to -\infty$ and hence $\Phi(q_i - |\boldsymbol{c}|\sqrt{N}) \to 0$. Consequently, the Type II error $\beta_i = 1 - P^i_{\text{detect}}$ decays to zero. Thus, for any fixed nonzero correlation magnitude $|\boldsymbol{c}| > 0$, sufficiently large N ensures arbitrarily high detection probability while maintaining controlled Type I error.

Algorithm 3 summarizes the maximum-statistic classification procedure with Bonferroni-controlled thresholds.

D.3 Data Complexity for Maximum Statistic Approach

The data complexity depends on the required detection probability and the number of tested assignments.

Theorem 6 (Data Complexity for Maximum Statistic Detection). *To achieve detection probability $P_{\text{detect}} = 1 - \beta$ for weak-key class $\mathcal{W}_i$ with s_i tested*

[2] This formula assumes the z_x are independent under H_1. As noted earlier, assignments share non-active nibble patterns, so strict independence does not hold. The formula provides a useful approximation when the number of non-active nibbles is large.

Algorithm 3: Maximum-Statistic Classification with FWER Control

Input: Encryption oracle E_K, input difference δ, output mask ω, weak-key classes $\{\mathcal{W}_1, \ldots, \mathcal{W}_\ell\}$ with assignment sets $\{\mathcal{X}_1, \ldots, \mathcal{X}_\ell\}$, samples per assignment N, FWER budget α, detection threshold θ

Output: Classification: `WEAK` or `STRONG`

`// Step 1: Collect data once and reuse it across all classes`

Query $D_{\text{total}} = N \times 2^{b \cdot m_{\text{in}}}$ plaintext-ciphertext pairs and store all pairs;

`// Step 2: Allocate per-class error rates and compute thresholds`

for $i \leftarrow 1$ **to** ℓ **do**

  $s_i \leftarrow |\mathcal{X}_i|$;

  $\alpha_i \leftarrow \alpha/\ell$; `// Bonferroni allocation;`

  $q_i \leftarrow \Phi^{-1}\left(1 - \frac{\alpha_i}{2s_i}\right)$; `// Threshold from Theorem 5`

`// Step 3: Evaluate the maximum statistic for each class`

$h \leftarrow 0$; `// Number of classes whose maxima exceed their thresholds`

for $i \leftarrow 1$ **to** ℓ **do**

  foreach $x \in \mathcal{X}_i$ **do**

    Extract the N stored pairs $(P_j, P_j \oplus \delta)$ whose active nibbles realize x;

    for $j \leftarrow 1$ **to** N **do**

      Compute $(C_j, C'_j) \leftarrow (E_K(P_j), E_K(P_j \oplus \delta))$;

      $B_{x,j} \leftarrow (-1)^{\omega \cdot (C_j \oplus C'_j)}$; `// DL distinguisher bit`

    $\widehat{\boldsymbol{c}}_x \leftarrow \frac{1}{N}\sum_{j=1}^{N} B_{x,j}$; `// Empirical autocorrelation`

    $z_x \leftarrow \sqrt{N} \cdot |\widehat{\boldsymbol{c}}_x|$; `// See Definition 17`

  $M_{s_i} \leftarrow \max_{x \in \mathcal{X}_i} z_x$;

  if $M_{s_i} \geq q_i$ **then**

    $h \leftarrow h + 1$;

`// Step 4: Final decision (count-based threshold from Section 4)`

if $h \geq \theta$ **then**

  return `WEAK` ; `// At least` θ `classes exhibit a decisive assignment`

return `STRONG`

assignments, autocorrelation magnitude $|\boldsymbol{c}|$, and Type I error α, the required number of samples per assignment is approximately

$$N \approx \left(\frac{q_{\alpha,s_i} - \Phi^{-1}(\beta^{1/s_i})}{|\boldsymbol{c}|} \right)^2,$$

where $q_{\alpha,s_i} = \Phi^{-1}(1 - \alpha/(2s_i))$ is the threshold from Theorem 5.

Proof. Under H_1, we require $\Pr(M_{s_i} > q \mid H_1) \geq 1 - \beta$. For the maximum of s_i absolute values $|z_x|$ where z_x has mean $\pm|\boldsymbol{c}|\sqrt{N}$ and variance 1, we have

$$\Pr(M_{s_i} \leq q \mid H_1) = [\Phi(q - |\boldsymbol{c}|\sqrt{N})]^{s_i}.$$

Setting $[\Phi(q - |\boldsymbol{c}|\sqrt{N})]^{s_i} = \beta$ and solving:

$$\Phi(q - |\boldsymbol{c}|\sqrt{N}) = \beta^{1/s_i} \quad \Rightarrow \quad q - |\boldsymbol{c}|\sqrt{N} = \Phi^{-1}(\beta^{1/s_i}).$$

Substituting $q = q_{\alpha,s_i}$ and solving for N yields the result.

Remark 5. The total data complexity is $D = N \times 2^{b \cdot m_{\text{in}}}$, where m_{in} is the number of active nibbles and b is the word size ($b = 4$ for Orthros). Only a fraction $s_i/2^{b \cdot m_{\text{in}}}$ of the queries correspond to good assignments for class $\mathcal{W}_i$.

Since different weak-key classes have different sizes $s_i = |\mathcal{X}_i|$, the required N to achieve target power varies by class. In practice, the algorithm uses a single value N for all classes, typically chosen as $N = \max_i N_i$ to ensure all classes meet the minimum detection probability, where N_i is computed from the theorem for each class size. Alternatively, N can be set based on the smallest class sizes if detecting those classes is most critical.

D.4 Application to 5-Round Orthros Distinguisher

We now apply the maximum statistic method to construct a 6-round universal attack on Orthros-PRF based on the 5-round distinguisher from Figure 5. This allows us to quantify its performance relative to the LLR-based method from Section 4. The distinguisher operates on rounds 2–6 of Orthros, and we prepend one round of key recovery (round 1) to mount a 6-round attack.

Threshold Calculation. Recall from Table 12 that the distinguisher achieves autocorrelation $\boldsymbol{c} = 2^{-11}$ for a single output mask (product of two independent single-branch correlations: $2^{-5} \times 2^{-6}$). From the 5-round analysis, the weak-key classes have the following assignment sizes:

- $|\mathcal{X}_i| = 2$: 576 classes
- $|\mathcal{X}_i| = 4$: 1152 classes
- $|\mathcal{X}_i| = 8$: 512 classes
- Total: $\ell = 2240$ weak-key classes

For FWER control at level $\alpha = 0.01$, we use Bonferroni correction: $\alpha_i = \alpha/\ell = 0.01/2240 = 4.46 \times 10^{-6}$ per class. The class-specific thresholds from Theorem 5 are:

- $|\mathcal{X}_i| = 2$: $q_2 = \Phi^{-1}(1 - 4.46 \times 10^{-6}/(2 \times 2)) = 4.7312$
- $|\mathcal{X}_i| = 4$: $q_4 = \Phi^{-1}(1 - 4.46 \times 10^{-6}/(2 \times 4)) = 4.8700$
- $|\mathcal{X}_i| = 8$: $q_8 = \Phi^{-1}(1 - 4.46 \times 10^{-6}/(2 \times 8)) = 5.0052$

Performance with Same Data as LLR ($N = 2^{20}$). With $\boldsymbol{c} = 2^{-11}$ and $N = 2^{20}$, we have $|\boldsymbol{c}|\sqrt{N} = 2^{-11} \times 2^{10} = 2^{-1} = 0.5$. Using the detection probability formula $P_{\text{detect}} = 1 - [\Phi(q_i - |\boldsymbol{c}|\sqrt{N})]^{s_i}$:

- $|\mathcal{X}_i| = 2$: $\Phi(4.7312 - 0.5) = \Phi(4.2312) \approx 0.999988$, $P_{\text{detect}} \approx 1 - (0.999988)^2 \approx 0.0024\%$
- $|\mathcal{X}_i| = 4$: $\Phi(4.8700 - 0.5) = \Phi(4.3700) \approx 0.999994$, $P_{\text{detect}} \approx 1 - (0.999994)^4 \approx 0.0025\%$
- $|\mathcal{X}_i| = 8$: $\Phi(5.0052 - 0.5) = \Phi(4.5052) \approx 0.999997$, $P_{\text{detect}} \approx 1 - (0.999997)^8 \approx 0.0027\%$

These are approximately 26000–30000× worse than LLR's 70% detection rate with the same $N = 2^{20}$.

The fundamental issue is that strict FWER control requires high thresholds $q_i \in [4.73, 5.01]$. With $|\boldsymbol{c}|\sqrt{N} = 0.5$, the gap $q_i - |\boldsymbol{c}|\sqrt{N} \in [4.23, 4.50]$ is large,

meaning each individual assignment has probability $\Phi(4.23) \approx 0.999988$ of *not* exceeding the threshold even under H_1 (weak-key). With only $s_i \in \{2, 4, 8\}$ independent tests per class, detection probability remains negligible. Additionally, the maximum statistic method uses only a single output mask rather than exploiting all 15 nonzero masks as the LLR method does, further reducing detection power.

Data Requirements for Target Detection Probabilities. Table 8 shows the data required for 70% and 95% detection rates.

Table 8: Data requirements and error analysis for maximum statistic method with $|\boldsymbol{c}| = 2^{-11}$ and FWER $\alpha = 0.01$.

Class size $\lvert\mathcal{X}_i\rvert$	**Target power**	N	D	**Type I error** (α_i)	**Type II error** (β)
For 70% Detection Rate ($\beta = 30\%$)					
2	70%	$2^{26.41}$	$2^{34.41}$	4.46×10^{-6}	30%
4	70%	$2^{26.16}$	$2^{34.16}$	4.46×10^{-6}	30%
8	70%	$2^{25.94}$	$2^{33.94}$	4.46×10^{-6}	30%
For 95% Detection Rate ($\beta = 5\%$)					
2	95%	$2^{26.91}$	$2^{34.91}$	4.46×10^{-6}	5%
4	95%	$2^{26.61}$	$2^{34.61}$	4.46×10^{-6}	5%
8	95%	$2^{26.35}$	$2^{34.35}$	4.46×10^{-6}	5%

Comparison with LLR. The LLR method achieves 70% detection with $N = 2^{20}$ ($D = 2^{28}$). For 70% detection, the maximum statistic requires **62–85× more data** ($D \in [2^{33.94}, 2^{34.41}]$ vs 2^{28}). For 95% detection, the gap widens further: the maximum statistic needs **99–120× more data** ($D \in [2^{34.35}, 2^{34.91}]$ vs approximately $2^{28.3}$ for LLR).

Error Analysis. The Type I error per class is controlled at $\alpha_i = \alpha/\ell = 0.01/2240 = 4.46 \times 10^{-6}$ via Bonferroni correction, ensuring FWER ≤ 0.01. The Type II error β decreases from 30% (70% power) to 5% (95% power) as N increases from $\approx 2^{26}$ to $\approx 2^{27}$. However, even with these data requirements, the maximum statistic remains fundamentally less efficient than LLR due to the threshold bottleneck: with only $s_i \in \{2, 4, 8\}$ independent tests per class, the method cannot fully exploit the signal strength.

D.5 Comparison with LLR-Based Method

Having seen the concrete performance on the 6-round Orthros attack, we now systematically compare the maximum statistic approach with the LLR-based method from Section 4.

Fundamental Differences. Table 9 summarizes the key differences between the two methods.

Table 9: Comparison of LLR and Maximum Statistic Classification Methods

Property	**LLR Method**	**Maximum Statistic Method**
Distinguisher Type	Multidimensional (m-bit output)	Single output mask
Signal Strength	Capacity $\mathrm{Cp}(F) = \sum_{a \neq 0} \mathrm{Cr}^2(a \cdot F)$	Single autocorrelation $\lvert\boldsymbol{c}\rvert$
Test Statistic	$\mathrm{LLR} = M_i \sum_y \widehat{p}_y \log \frac{p_y}{q_y}$	$M_{s_i} = \max_{x \in \mathcal{X}_i} \lvert z_x \rvert$
Data Complexity	$N \approx d/\mathrm{Cp}(F)$, then $D = N \times 2^{b \cdot m_{\mathrm{in}}}$	$N_i \approx \left\lceil \frac{q_i - \Phi^{-1}(\beta^{1/s_i})}{\lvert\boldsymbol{c}\rvert} \right\rceil^2$, use $N = \max_i N_i$, $D = N \times 2^{b \cdot m_{\mathrm{in}}}$
How Class i Uses Data	Pools $M_i = \lvert\mathcal{X}_i\rvert \times N$ pairs into one statistic	Computes $s_i = \lvert\mathcal{X}_i\rvert$ separate statistics of N pairs each
Null Distribution	$\mathcal{N}(M_i\mu_1, M_i\sigma_1^2)$	Maximum of s_i i.i.d. half-normals
Alternative Dist.	$\mathcal{N}(M_i\mu_0, M_i\sigma_0^2)$	Maximum of s_i i.i.d. folded normals
Threshold	$\tau_i = M_i\mu_0 - \sqrt{M_i}\sigma_0\Phi^{-1}(P_{\mathrm{detect}})$	$q_i = \Phi^{-1}(1 - \alpha_i/(2s_i))$
Optimality	Neyman-Pearson optimal	Conservative (union bound approximation)
FWER Control	Via threshold adjustment	Rigorous via Bonferroni correction

Distinguisher Structure. A fundamental architectural difference is that the LLR method exploits a multidimensional DL distinguisher using all nonzero output masks simultaneously (e.g., all 15 masks for a 4-bit output space), leveraging capacity $\mathrm{Cp}(F(X)) = \sum_{a \in \mathbb{F}_2^m \setminus \{0\}} \mathrm{Cr}^2(a \cdot F)$. In contrast, the maximum statistic uses a single DL distinguisher with one output mask, working with a single autocorrelation value $\lvert\boldsymbol{c}\rvert$. This design difference significantly impacts data efficiency: the multidimensional approach gains roughly a factor of 2^m in sample complexity when all correlations are comparable.

Statistical Optimality. The LLR approach aggregates information from all $M_i = \lvert\mathcal{X}_i\rvert \times N$ samples. By the Neyman-Pearson lemma (Lemma 2), the LLR test is most powerful for a given Type I error rate when testing a single hypothesis. The maximum statistic is inherently conservative, requiring only that at least one assignment exceeds the threshold rather than aggregate evidence from all assignments.

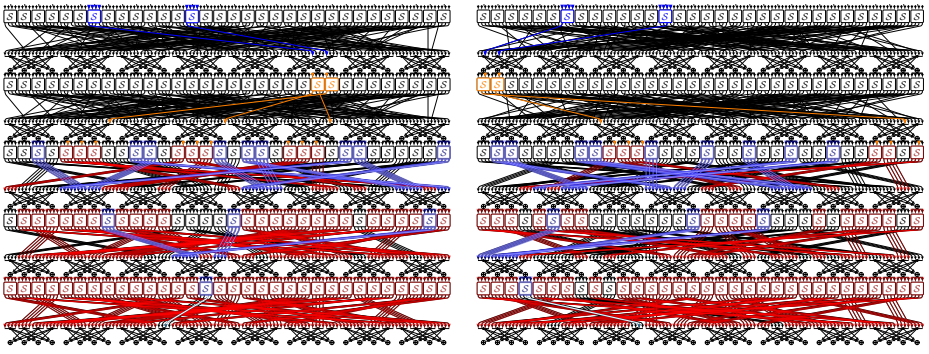
E Empirical Correlations for Strong vs. Weak-Keys

This appendix reports empirical distributions of the estimated correlations for strong and weak-keys in our 5- and 6-round DL attacks on Orthros-PRF. We define the standardized statistic we use, explain how we plot it, and show how to read the figures.

Per-test statistic and z-score. For a fixed distinguisher and a fixed setting of the active positions we evaluate N samples and record the DL output bit $B_j \in \{+1, -1\}$ on sample j. We define the empirical correlation as $\widehat{\boldsymbol{c}} \coloneqq \frac{1}{N}\sum_{j=1}^{N} B_j$ and drop any class or key subscripts here for readability. We define the raw correlation count as $C \coloneqq \left|\sum_{j=1}^{N} B_j\right| = N \cdot |\widehat{\boldsymbol{c}}|$. Under the strong-key null the sum $\sum_{j=1}^{N} B_j$ is approximately normal with mean 0 and variance N so $C/\sqrt{N}$ is approximately half-normal. We standardize via $z \coloneqq C/\sqrt{N} = \sqrt{N} \cdot |\widehat{\boldsymbol{c}}|$. Horizontal reference lines at $z \in \{2.5, 3, 4\}$ correspond to count levels $C \in \{2.5, 3, 4\}\sqrt{N}$. We use these fixed z thresholds to compare across different N.

E.1 Five-Round DL Attack on Orthros-PRF

Figure 14 shows the 5-round key-recovery structure with `Offset` $= 2$. The attack adds one round of key recovery and uses a 4-round DL distinguisher. The DL part decomposes into a 1-round differential distinguisher followed by a 3-round combined DL distinguisher. Since the deterministic differential and linear trails do not overlap in the combined part, its correlation equals 1. Let $p_1 = p_2 = 2^{-4}$ denote the 1-round differential probability in Branch1 and Branch2, respectively. The distinguisher requires both events, so $p = p_1 p_2 = 2^{-8}$ and the 4-round DL correlation is $\mathrm{Cr} \approx 2^{-8}$. This suggests that we need about $N \gtrsim 2^{16}$ samples per good pair to observe the correlation reliably.



(a) Branch1 of 6-round Orthros (b) Branch2 of 6-round Orthros

Fig. 14: 5-round DL attack used for the empirical study.

With $\texttt{Offset} = 2$, the involved key-bit indices are $[45, 114, 98, 69, 41, 73, 80, 79]$ in Branch1 and $[87, 125, 89, 78, 37, 101, 60, 127]$ in Branch2 as indicated in Figure 14. There are 2^{16} assignments for these bits, but only 12800 are weak-keys, which is about $2^{-2.36}$ of the space. The number of good assignments for the active nibbles in the key recovery part is $2304 \approx 2^{11.17}$. For each good assignment, we tested $N \in \{2^{16}, 2^{17}, 2^{18}, 2^{19}\}$ plaintext pairs with random choices for the inactive positions. We refer to the resulting multi-panel plot for each setting as the *signature* of the key candidate.

How to read the signature figures Each signature has four panels that summarize the same experiment from complementary angles. We explain how to interpret the results in each panel. The bar and scatter panels use a linear vertical scale so relative magnitudes of large counts remain visually proportional to their actual excess above noise. The histogram uses a logarithmic y-axis so that rare large counts in the right tail remain visible without flattening the body of the distribution. When comparing two signatures focus first on the bar and scatter z exceedances and then read the histogram tail mass for corroboration.

Bar panel. We plot the raw counts C for all tests on a linear vertical axis and draw horizontal reference lines for the noise mean, the 2.5σ and 3σ thresholds, and an adaptive family-wise error rate (FWER) threshold at $z = q_{\text{ext}}$ (typically $\geq 4\sigma$). Bars with $z \geq 3$ receive a bold outline and colors encode the z tier (< 2.5, $[2.5, 3)$, $[3, 4)$, $[4, q_{\text{ext}})$, $\geq q_{\text{ext}}$). A translucent red glow highlights the region above 4σ, and the legend reminds the reader that $z = C/\sqrt{N}$ so counts can be compared across datasets with different sample sizes. Exceedances above the FWER line are individually decisive under the strong-key null.

Peak spectrum. We sort tests by decreasing C and plot at most the top 50 peaks. We overlay a log–log least-squares power-law fit (dashed) whose slope quantifies the decay rate. Weak keys start higher and the early ranks often deviate sharply above the fitted decay implying a handful of dominant peaks. Strong keys start lower and follow a smoother, more gradually decaying profile.

Scatter view. We show the cloud of points (color-coded by z tier) and the running maximum on a linear C axis. Point size increases at 3σ, 4σ, and the FWER threshold; true good pairs (when known) appear as gold star markers; and peaks with $z \geq 3$ are ring-highlighted. The running maximum trajectory shows when decisive evidence first appears and how sustainably it pulls away from noise.

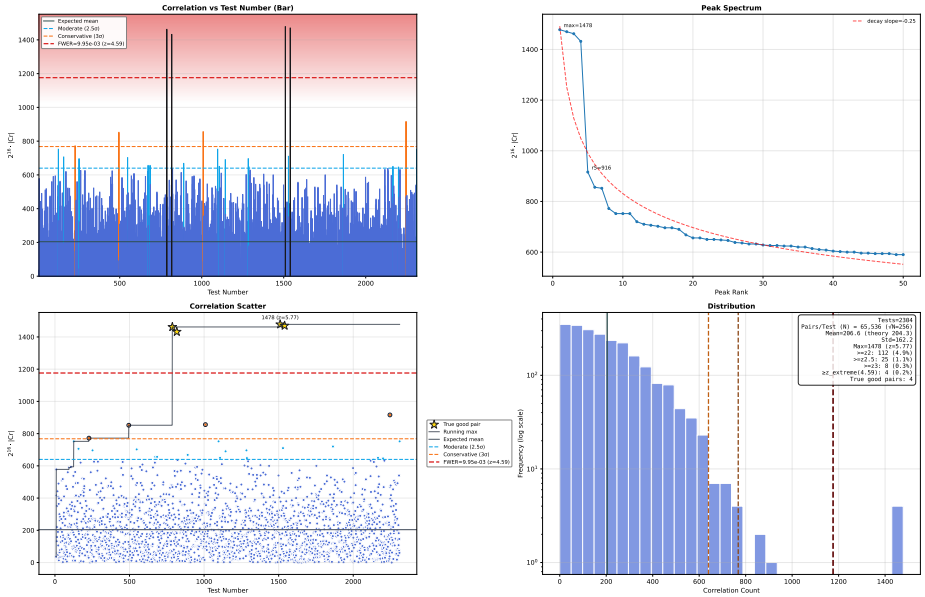
Histogram. We plot the distribution of C (density) with a logarithmic y-axis so rare large counts remain visible. Vertical lines mark the same thresholds (mean, 2.5σ, 3σ, FWER) even though only the latter three drive detection. Counts are nonnegative so the baseline shape under a strong-key approximates a folded normal tail near zero. A weak-key produces extra mass in the upper tail with multiple bins beyond 3σ and often at least one spike beyond the FWER line. An inset summary (in the figure) enumerates exceedances at 2σ, 2.5σ, 3σ, and FWER (the 2σ level is reported but not drawn to avoid clutter).

Interpreting exceedance counts. Under the strong-key null each test exceeds a one-sided threshold $z^\star$ with probability $2(1-\Phi(z^\star))$, so with T tests we expect about $T \cdot 2(1-\Phi(z^\star))$ exceedances. Dependence among tests inflates variance so observed counts may deviate but the order of magnitude and weak vs strong separation persist. The adaptive FWER threshold q_{ext} is calibrated so that under the strong-key null the probability any test exceeds it is approximately controlled, making a single exceedance at that level strong evidence of a weak-key.

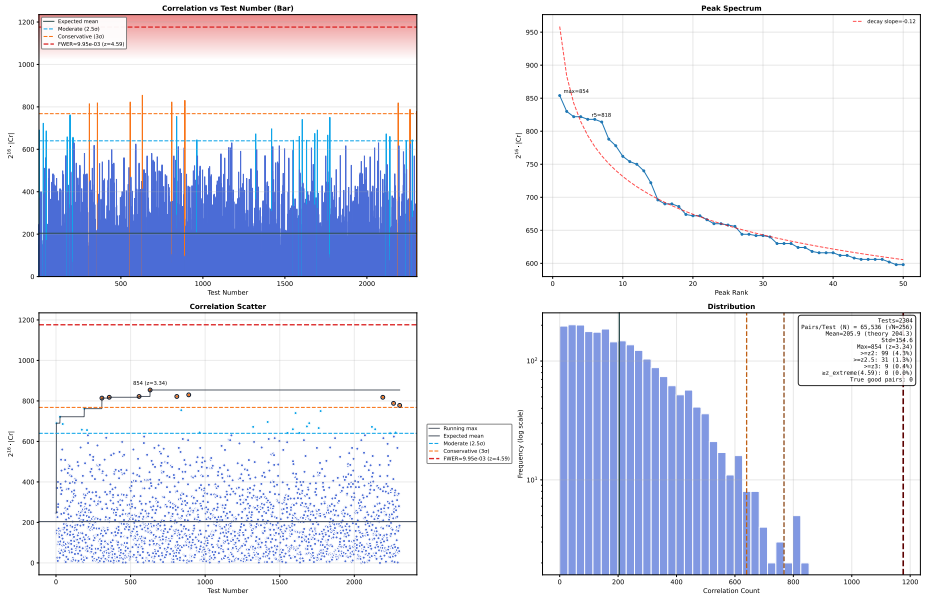
Main observations across N. Referring to Figure 15a, we can see that for $N = 2^{16}$ the weak-key signature already shows a heavy right tail, while the strong-key maximum stays near the conservative line. As N grows to 2^{17} and 2^{18} (see Figure 16a and Figure 17a) the spread contracts at a rate close to $1/\sqrt{N}$ and the overlap between weak and strong tails shrinks. As seen in Figure 18a, at $N = 2^{19}$ the separation is clear and a single threshold yields high power with low false-alarm rate. These observations match the prediction that $z = \sqrt{N}|\widehat{\boldsymbol{c}}|$ grows like $|\boldsymbol{c}|\sqrt{N}$ for a weak-key and remains near noise for a strong-key.

E.2 Six-Round DL Attack on Orthros-PRF

We repeated the experiment for the 6-round attack in Figure 5 with $\texttt{Offset} = 2$. The number of good pairs is $1152 \approx 2^{10.17}$. The 5-round DL distinguisher used inside the attack has correlation Cr $\approx 2^{-11}$, which suggests $N \approx 2^{22}$ samples per good pair. We measured two larger values, $N = 2^{27}$ and $N = 2^{28}$, to reduce overlap between weak and strong tails. For each N, the total number of cipher calls is about $2^{10.17} \cdot N$ per key, plus light processing to compute the empirical correlation. We implemented the evaluation in CUDA and ran it on an NVIDIA GeForce RTX 4090 to parallelize the computation.

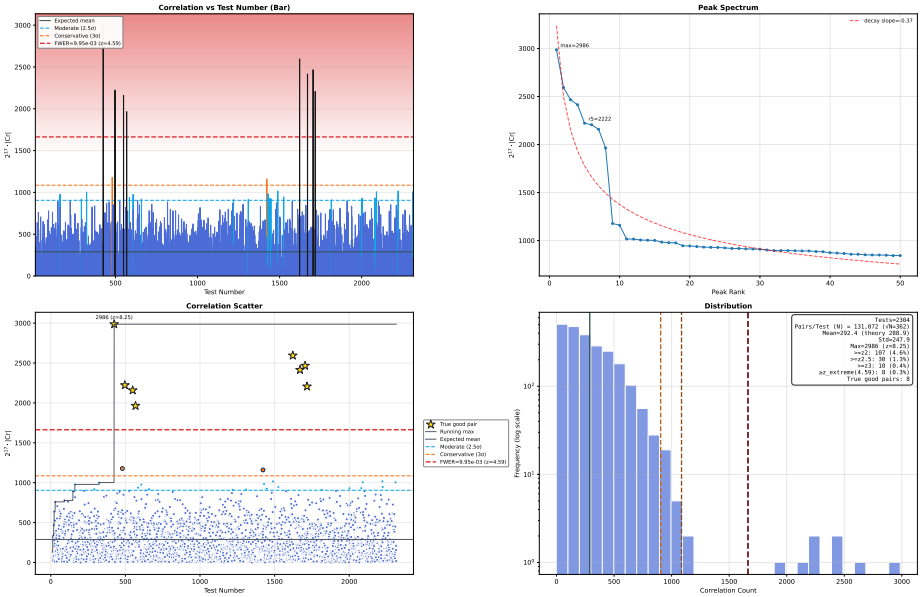


(a) Weak key.

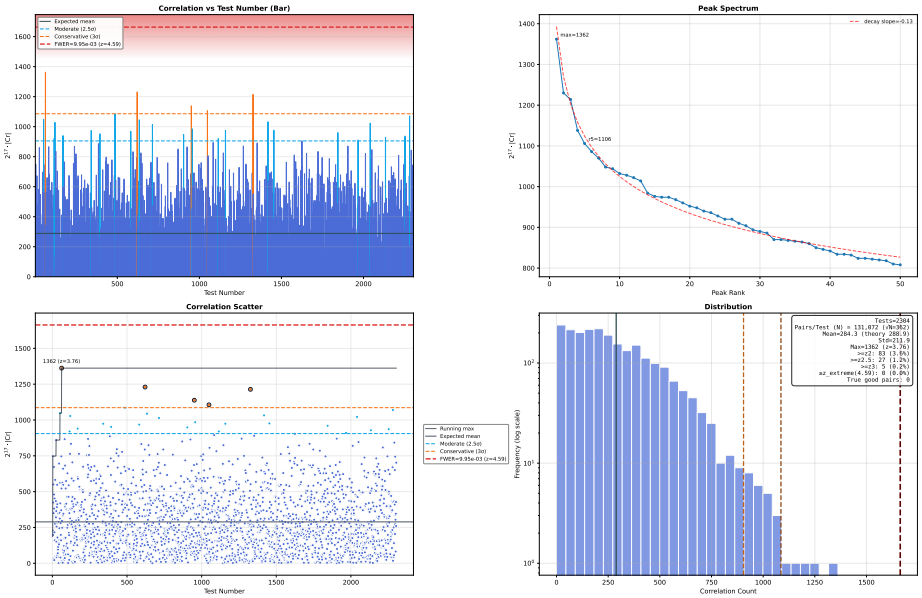


(b) Strong key.

Fig. 15: Distribution of empirical correlations for the 5-round attack in Figure 14 with $N = 2^{16}$ samples per good pair, shown for one weak-key and one strong-key.

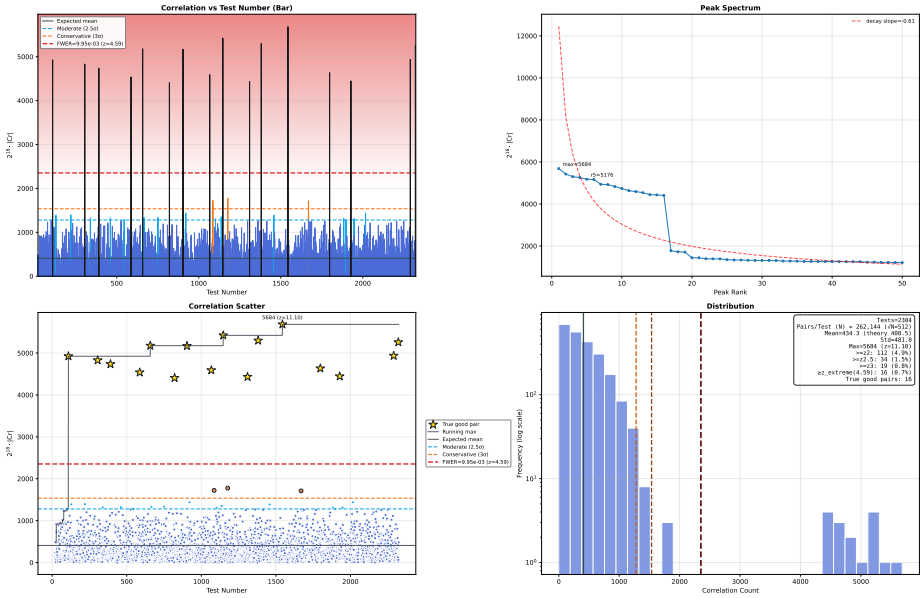


(a) Weak key.

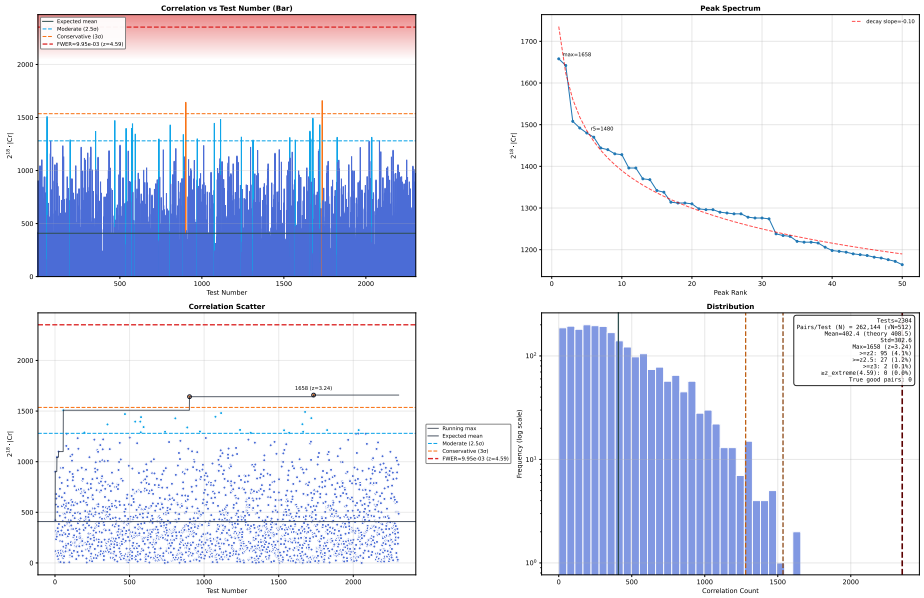


(b) Strong key.

Fig. 16: Distribution of empirical correlations for the 5-round attack in Figure 14 with $N = 2^{17}$ samples per good pair, shown for one weak-key and one strong-key.

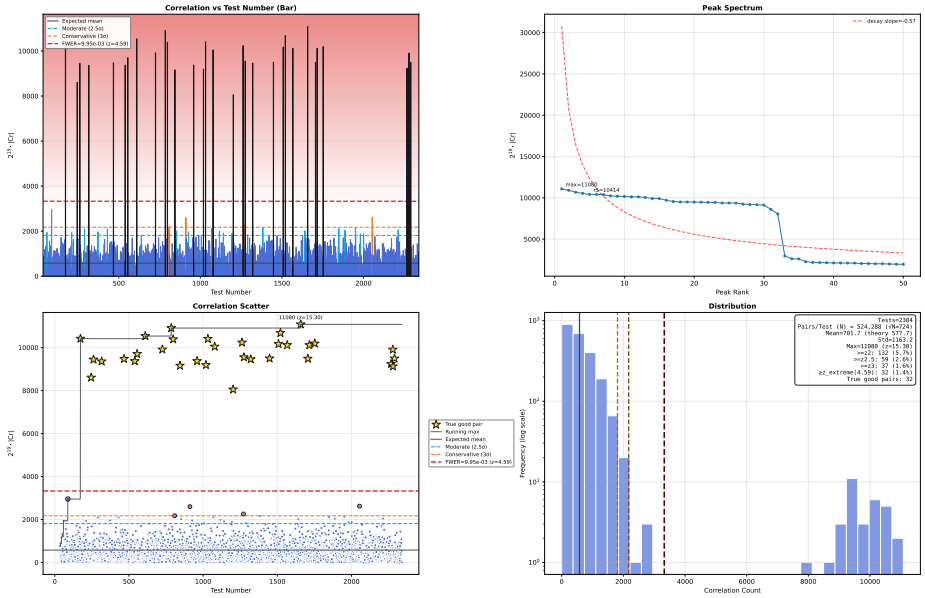


(a) Weak key.

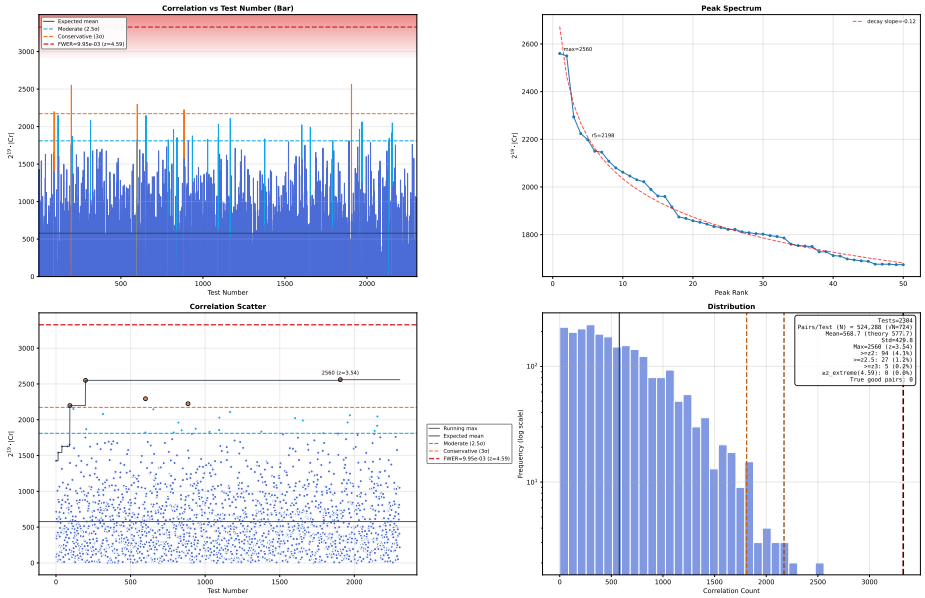


(b) Strong key.

Fig. 17: Distribution of empirical correlations for the 5-round attack in Figure 14 with $N = 2^{18}$ samples per good pair, shown for one weak-key and one strong-key.

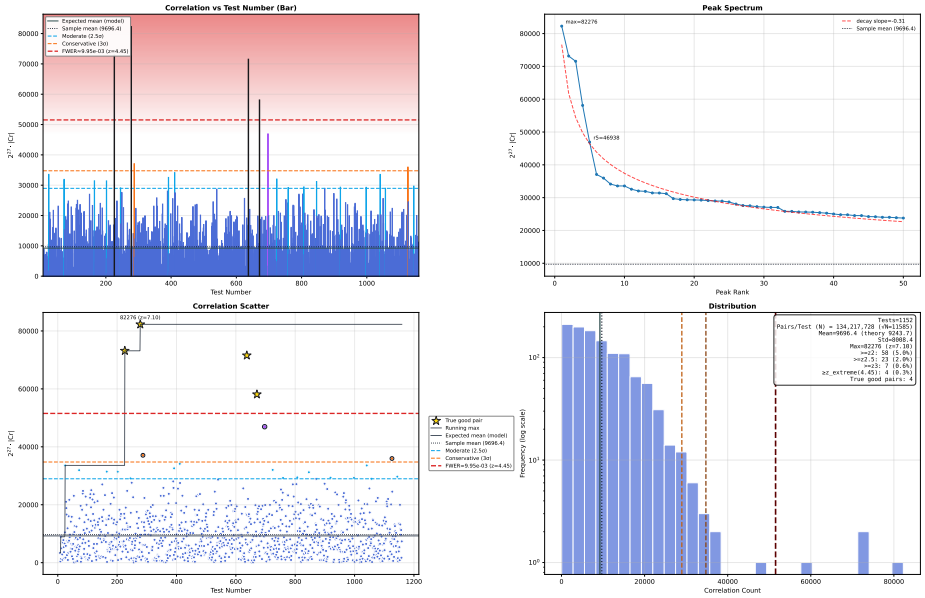


(a) Weak key.

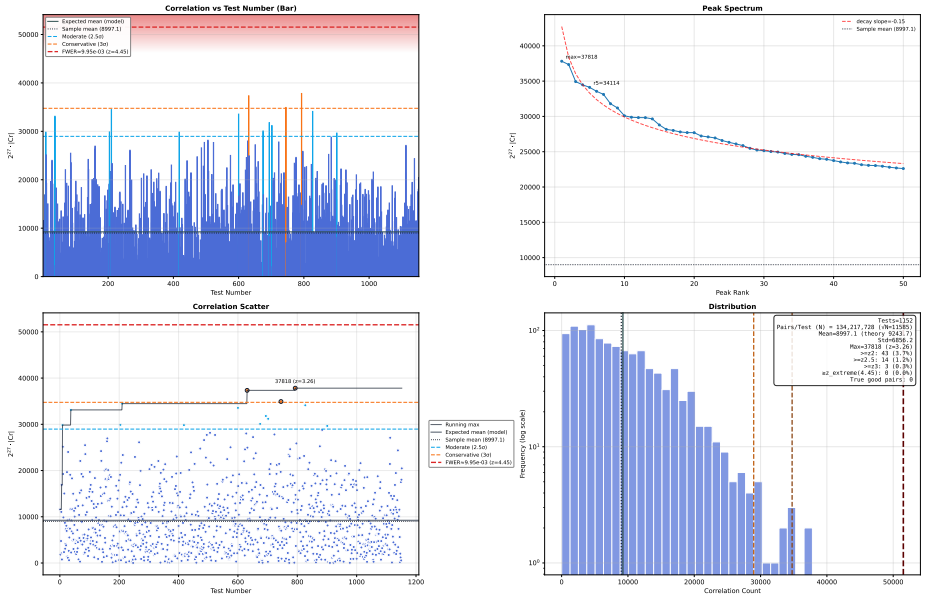


(b) Strong key.

Fig. 18: Distribution of empirical correlations for the 5-round attack in Figure 14 with $N = 2^{19}$ samples per good pair, shown for one weak-key and one strong-key.

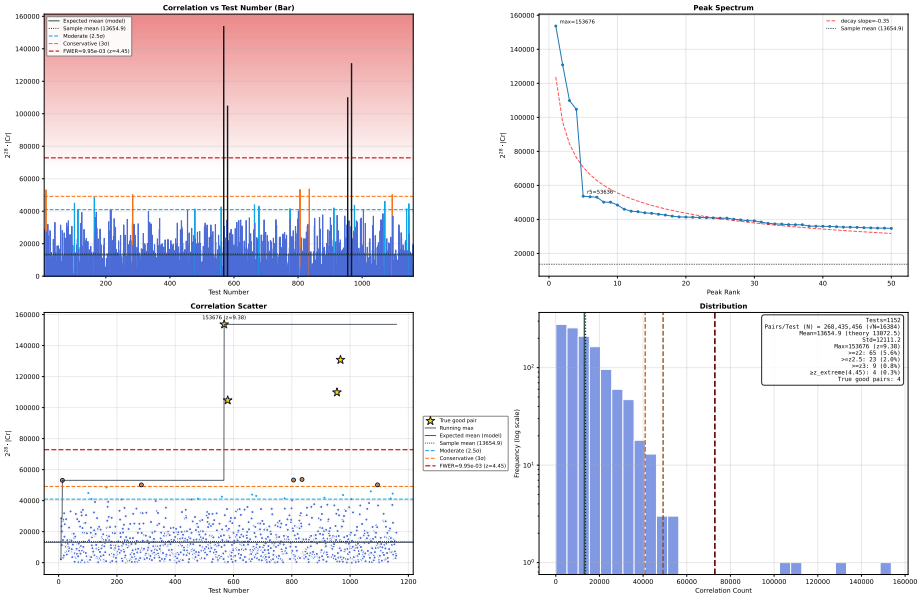


(a) Weak key.

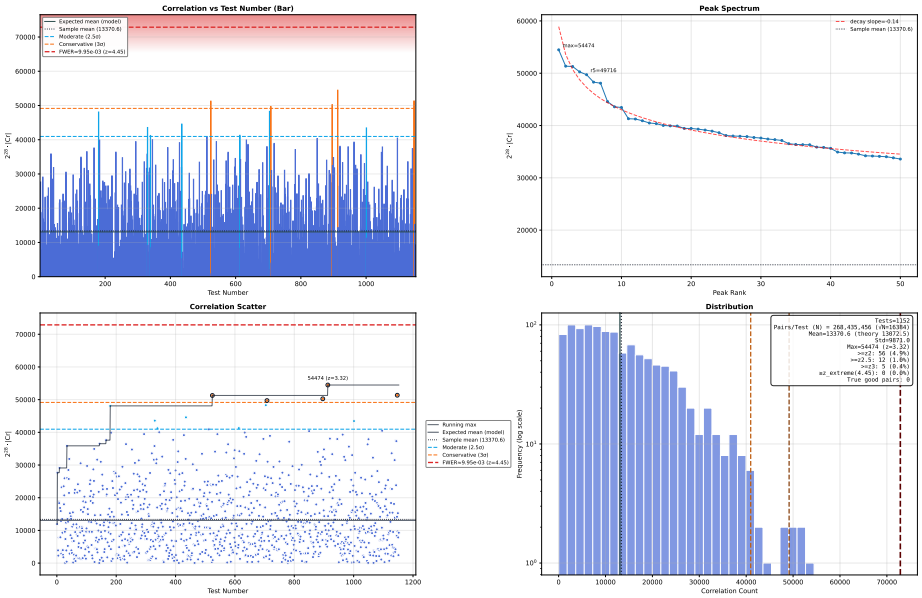


(b) Strong key.

Fig. 19: Distribution of empirical correlations for the 6-round attack in Figure 5 with $N = 2^{27}$ samples per good pair, shown for one weak-key and one strong-key.



(a) Weak key.



(b) Strong key.

Fig. 20: Distribution of empirical correlations for the 6-round attack in Figure 5 with $N = 2^{28}$ samples per good pair, shown for one weak-key and one strong-key.

F DL Distinguishers for Orthros Branch1

Let δ_i and ω_i denote the input difference and linear mask at round i, where i is counted starting from 0. Table 10 presents the DL distinguishers for `Branch1` of Orthros. In cases where the correlation is high enough to allow experimental verification, we also report the empirical correlation $\widehat{\mathrm{Cr}}(\delta_i, \omega_j)$.

Table 10: DL distinguishers for `Branch1` of Orthros.

5 Rounds, Figure 21	
$\texttt{Offset} = 4, r_{\mathrm{U}} = 0, r_{\mathrm{M}} = 5, r_\ell = 0,\ p = 1,\ r = 1,\ q^2 = 1,\ prq^2 = \mathbf{1}, \widehat{\mathrm{Cr}}(\delta_{\mathbf{0}}, \omega_{\mathbf{5}}) = \mathbf{1}$	
δ_0 : 00000000001000000000000000000000	ω_5 : 00000000000000000000101100000000
6 Rounds, Figure 22	
$\texttt{Offset} = 4, r_{\mathrm{U}} = 0, r_{\mathrm{M}} = 5, r_\ell = 1,\ p = 1,\ r = 1,\ q^2 = 2^{-6},\ prq^2 = 2^{-6}, \widehat{\mathrm{Cr}}(\delta_0, \omega_6) = 2^{-4.90}$	
δ_0 : 70000000000000800000000000c00000	
ω_5 : 0000000000000000ee0e000000000000	ω_6 : 0000000000000ccc0000000000ccc0ccc
9 Rounds, Figure 23	
$\texttt{Offset} = 0, r_{\mathrm{U}} = 4, r_{\mathrm{M}} = 4, r_\ell = 1,\ p = 2^{-21},\ r = 1,\ q^2 = 2^{-6},\ prq^2 = 2^{-27}$	
δ_0 : 00000000000000000000000800080000	δ_4 : 00000000998100000000000000000000
ω_8 : 00000000000000001101000000000000	ω_9 : 00000000000011100000000001110111
10 Rounds, Figure 24	
$\texttt{Offset} = 0, r_{\mathrm{U}} = 4, r_{\mathrm{M}} = 4, r_\ell = 2,\ p = 2^{-25},\ r = 2^{-7.75},\ q^2 = 2^{-8},\ prq^2 = 2^{-40.75}$	
δ_0 : 00000020000000000020000040000500	δ_4 : 00000000000000000000000080800000
ω_8 : 00000000000000010000000000000000	ω_{10} : 00000000000011100000000001110111

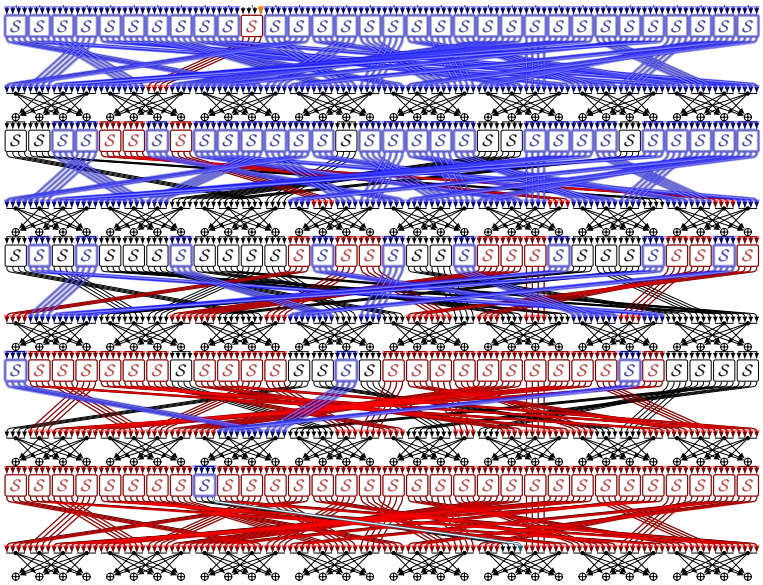


Fig. 21: DL distinguisher for 5 rounds of Branch1 in Orthros

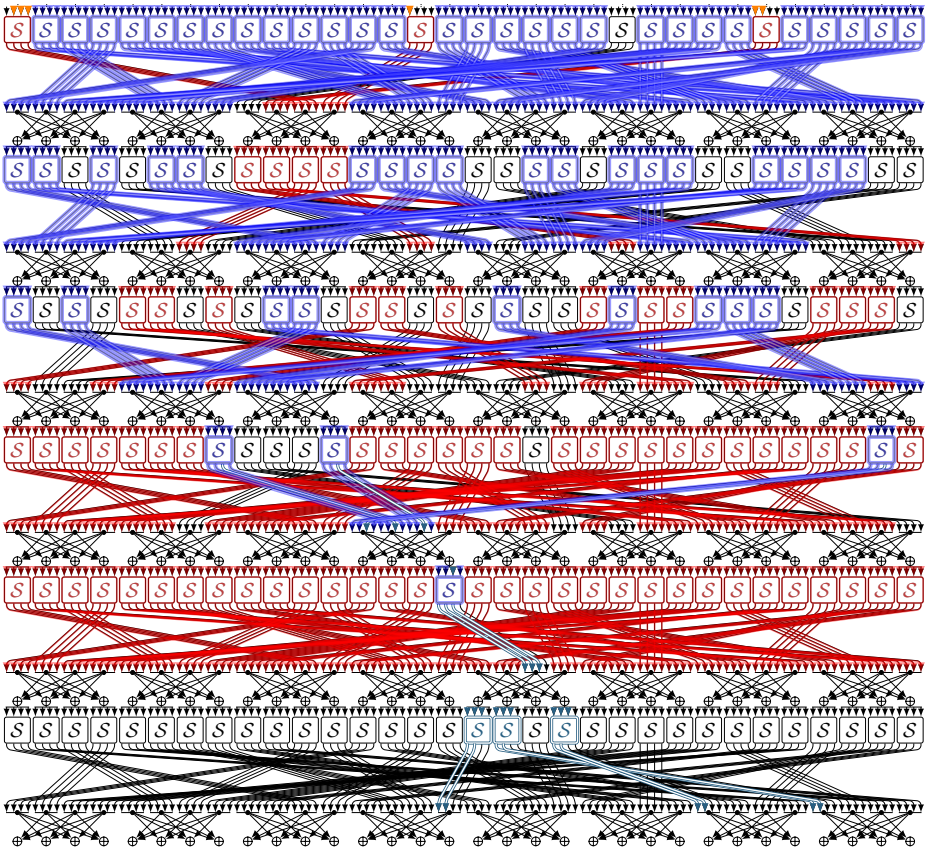


Fig. 22: DL distinguisher for 6 rounds of Branch1 in Orthros

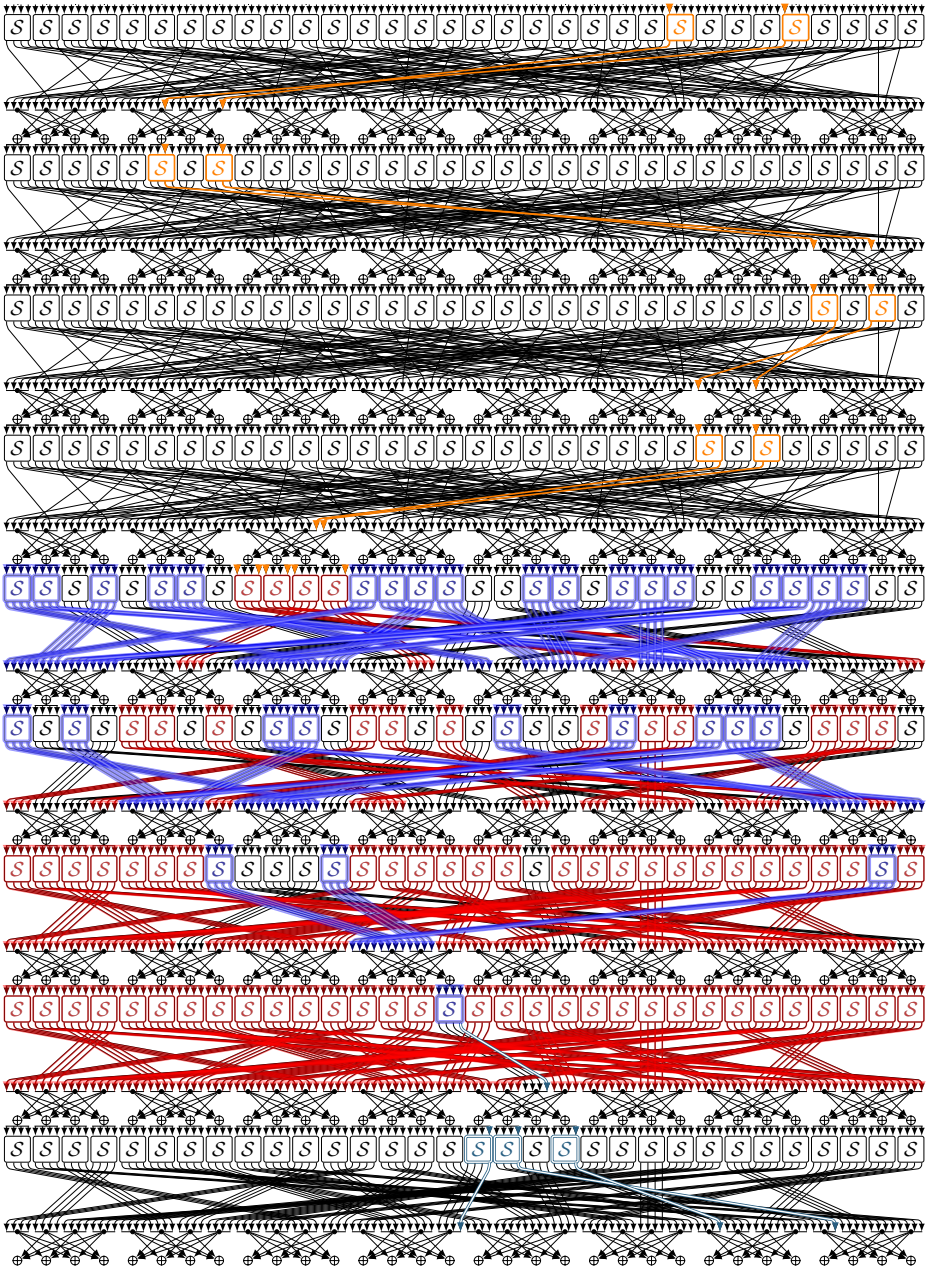


Fig. 23: DL distinguisher for 9 rounds of Branch1 in Orthros

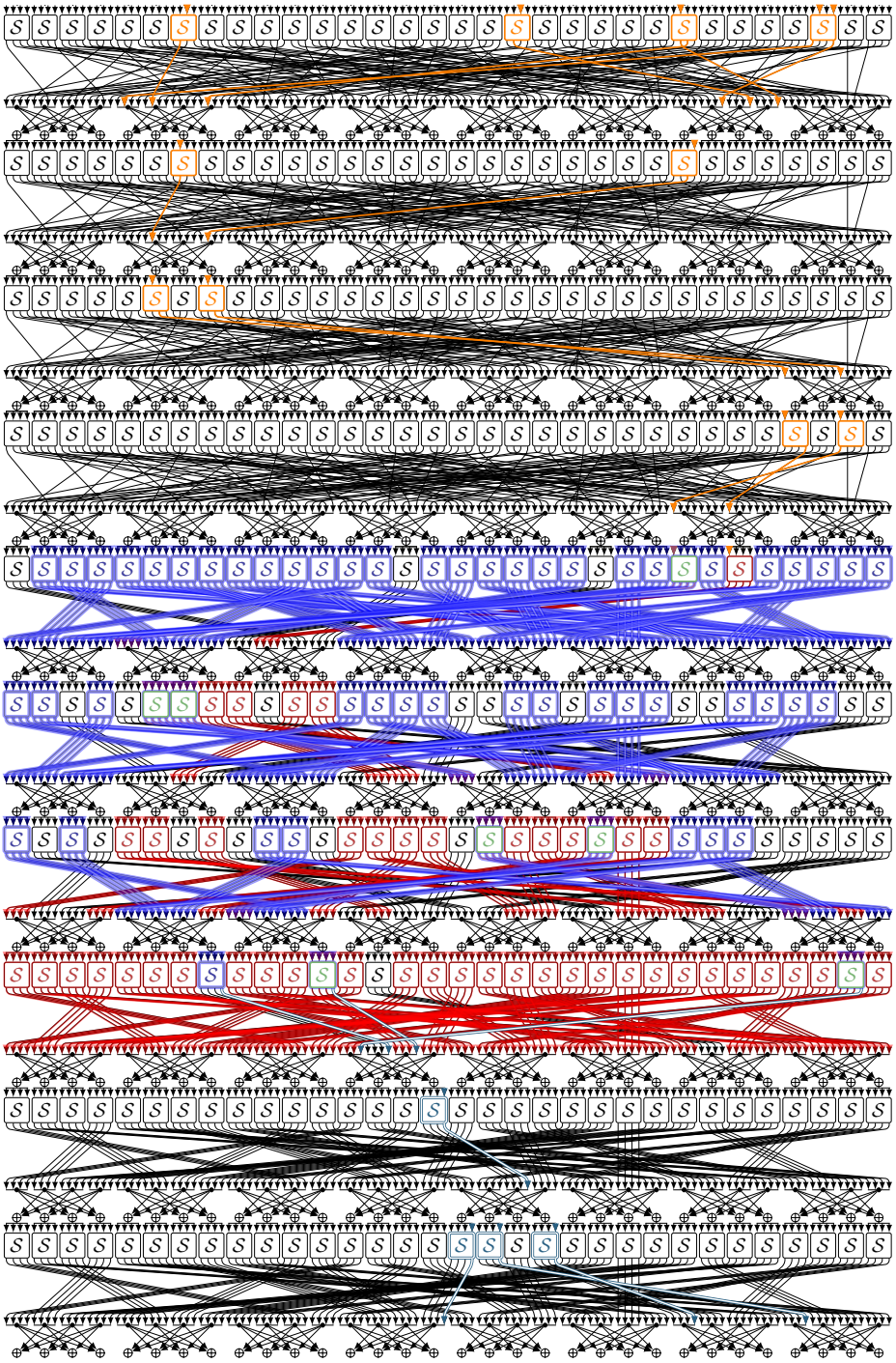


Fig. 24: DL distinguisher for 10 rounds of Branch1 in Orthros

G DL Distinguishers for Orthros Branch2

Let δ_i and ω_i denote the input difference and linear mask at round i, where i is counted starting from 0. Table 11 presents the DL distinguishers for `Branch2` of Orthros. In cases where the correlation is high enough to allow experimental verification, we also report the empirical correlation $\widehat{\mathrm{Cr}}(\delta_i, \omega_j)$.

Table 11: DL distinguishers for `Branch2` of Orthros.

5 Rounds, Figure 25 $\texttt{Offset} = 4, r_{\mathrm{U}} = 0, r_{\mathrm{M}} = 5, r_\ell = 0,\ p = 1,\ r = 1,\ q^2 = 1,\ prq^2 = \mathbf{1}, \widehat{\mathrm{Cr}}(\delta_{\mathbf{0}}, \omega_{\mathbf{5}}) = \mathbf{1}$	
δ_0 : 00040000000000000000000000000000	ω_5 : 88800000000000000000000000000000
6 Rounds, Figure 26 $\texttt{Offset} = 4, r_{\mathrm{U}} = 0, r_{\mathrm{M}} = 5, r_\ell = 1,\ p = 1,\ r = 1,\ q^2 = -6,\ prq^2 = 2^{-6}, \widehat{\mathrm{Cr}}(\delta_0, \omega_6) = 2^{-5.48}$	
δ_0 : 00000000000000400000000000000000 ω_5 : 00001011000000000000000000000000	ω_6 : 00000000000000002022101100002022
9 Rounds, Figure 27 $\texttt{Offset} = 0, r_{\mathrm{U}} = 4, r_{\mathrm{M}} = 4, r_\ell = 1,\ p = 2^{-22},\ r = 1,\ q^2 = 2^{-6},\ prq^2 = 2^{-28}$	
δ_0 : 00100000060000000000000000000000 ω_8 : 00002220000000000000000000000000	δ_4 : 00000000000000008008000000000000 ω_9 : 04440000000000004044000000004044
10 Rounds, Figure 28 $\texttt{Offset} = 0, r_{\mathrm{U}} = 4, r_{\mathrm{M}} = 4, r_\ell = 2,\ p = 2^{-23},\ r = 2^{-6.27},\ q^2 = 2^{-8},\ prq^2 = 2^{-37.27}$	
δ_0 : 02000000000000000000000000001000 ω_8 : 00000000000000000001000000000000	δ_4 : 00000000101000000000000000000000 ω_{10} : 10110000011100000000000000001110

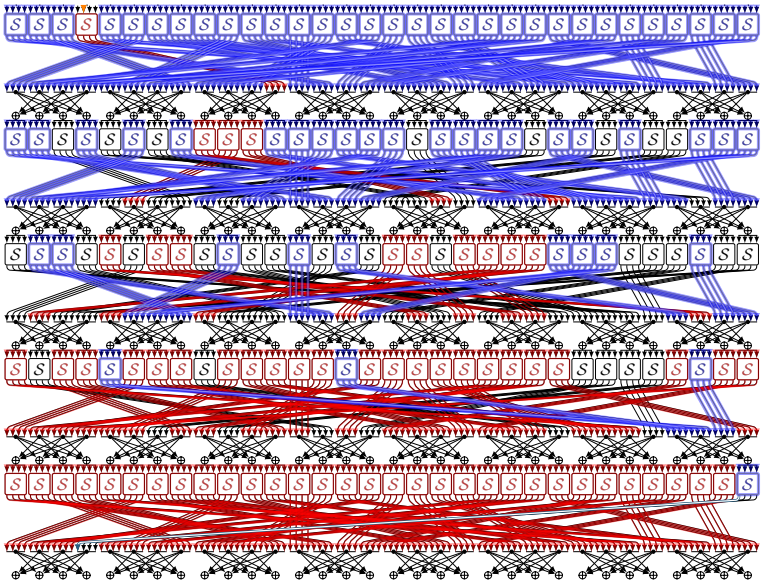


Fig. 25: DL distinguisher for 5 rounds of Branch2 in Orthros

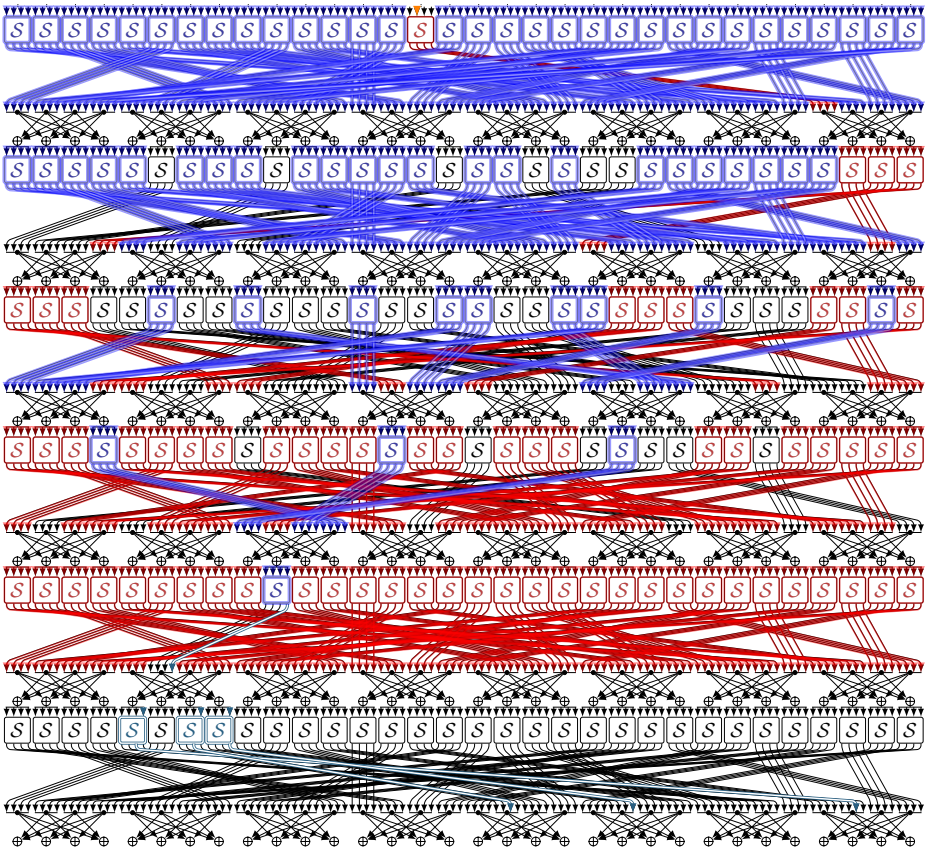


Fig. 26: DL distinguisher for 6 rounds of Branch2 in Orthros

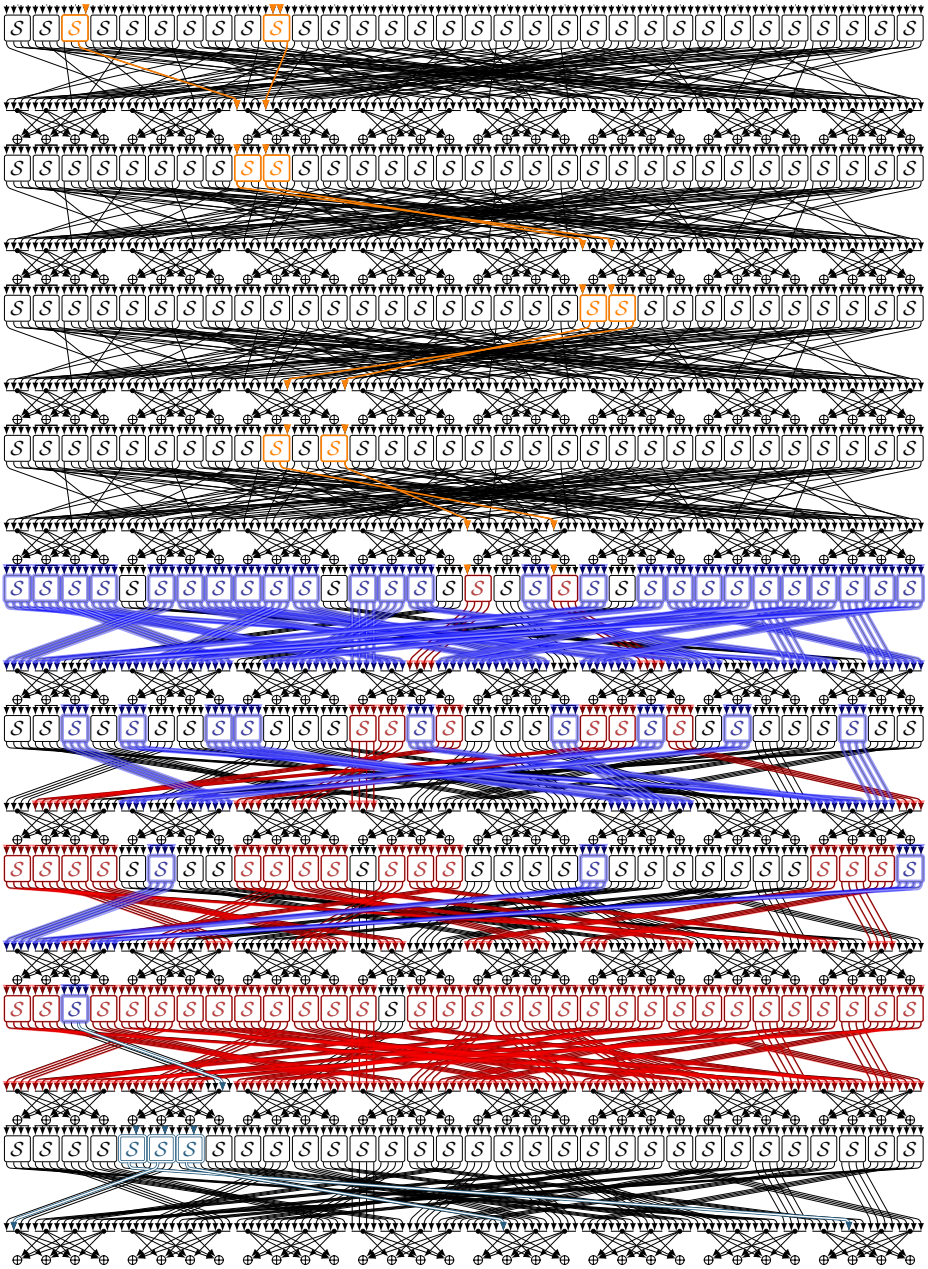


Fig. 27: DL distinguisher for 9 rounds of Branch2 in Orthros

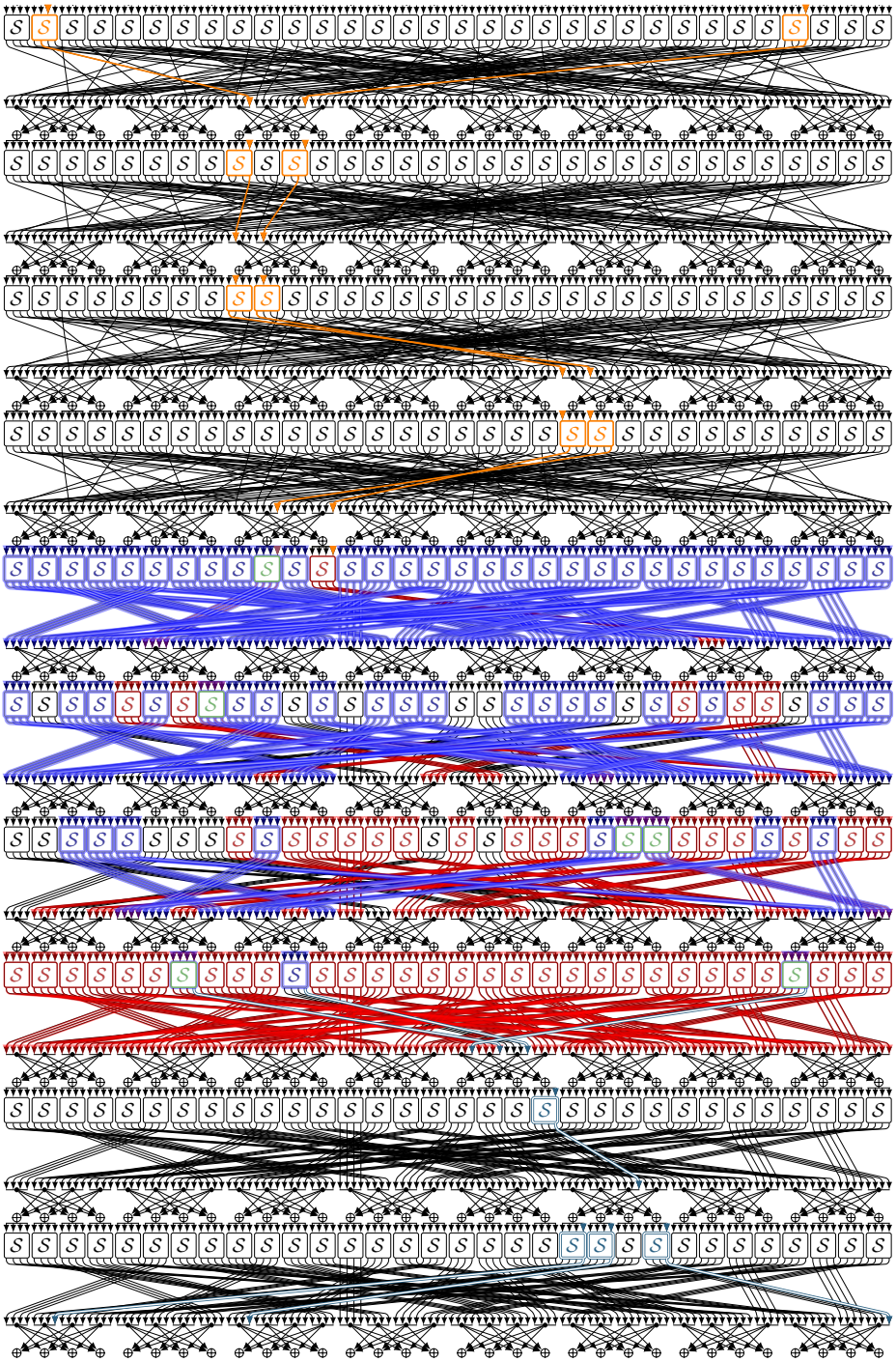


Fig. 28: DL distinguisher for 10 rounds of Branch2 in Orthros

H DL Attacks for Orthros PRF

Here, we present the DL distinguishers used in our key recovery attacks on Orthros-PRF. In this section, the number of rounds added for key recovery before the distinguisher is denoted by r_{B}.

Figure 5, Figure 31, Figure 32, and Figure 33 illustrate different distinguishers (and corresponding key recovery attacks) for 5-round (resp. 6-round) versions of Orthros-PRF. In these figures, the S-boxes and key bits involved in the key recovery part at the top are highlighted in blue. We follow the same notations for our 8-round attacks that are represented in Figure 34, Figure 35, Figure 37, and Figure 36.

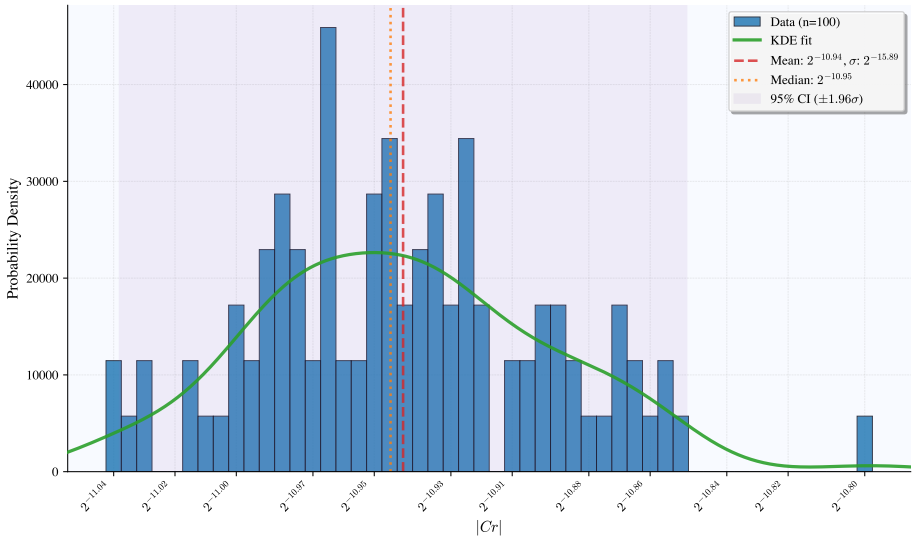


Fig. 29: Empirical correlation distributions for the 5-round distinguisher in Figure 5 (Orthros-PRF), evaluated over 100 random keys with $N = 2^{32}$ samples per key.

To verify that our estimation for the correlation of the SoP structure ($\mathrm{Cr}_t \approx \mathrm{Cr}_{\texttt{Branch1}} \cdot \mathrm{Cr}_{\texttt{Branch2}}$) holds, we performed extensive GPU experiments for the distinguishers marked by (✔💻) in Table 12. In each test, we selected a key at random, generated $N \in \{2^{32}, 2^{33}\}$ chosen-plaintext pairs, and computed the correlation of the DL distinguisher. This process was repeated for 100 independent random keys.

Since these distinguishers are used in key recovery after one round, we treat the two branches as statistically independent, assuming that the constraints on the plaintext and key bits are satisfied. This assumption is natural, as the main dependency is already handled by enforcing these constraints. Moreover,

we previously verified it in our experiments on the distributions of empirical correlations for strong and weak-keys in Section E.1 and Section E.2 (see, e.g., Figure 15). Accordingly, in our experiments to confirm the correlation of the SoP structure, we generated the input pairs of each branch independently to simulate independence.

Figure 29 reports results for the 5-round DL distinguisher 0 in Figure 5, evaluated over 100 random keys with $N = 2^{32}$ samples per key. The empirical correlations are consistent with the theoretical estimate 2^{-11}; the sample mean is $2^{-10.94}$ and the standard deviation is $2^{-15.89}$. Figure 30 reports results for the 5-round DL distinguisher 1 in Figure 31, evaluated over 100 random keys with $N = 2^{33}$ samples per key. The empirical correlations are consistent with the theoretical estimate 2^{-15}; the sample mean is $2^{-14.59}$ and the standard deviation is $2^{-16.49}$.

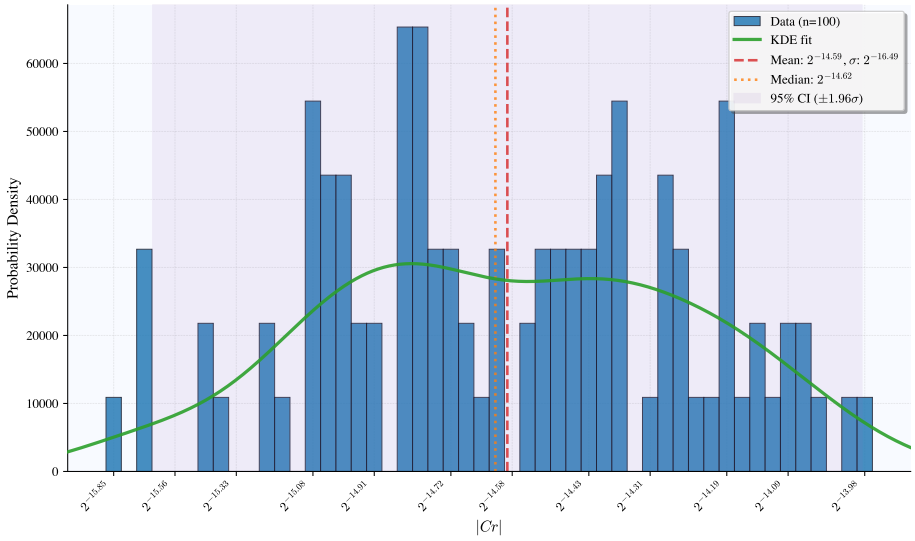


Fig. 30: Empirical correlation distributions for the 5-round distinguisher in Figure 31 (Orthros-PRF), evaluated over 100 random keys with $N = 2^{33}$ samples per key.

Table 12: DL distinguishers for 5 rounds of Orthros-PRF.

5-Round Distinguisher 0, Figure 5 ✔💻

$\texttt{Offset} = 2, r_{\text{B}} = 1, r_{\text{U}} = 1, r_{\text{M}} = 4, r_\ell = 0,\ p = 2^{-5},\ r[\texttt{x}] = 1,\ q^2 = 1, \text{Cr}[\texttt{x}] = prq^2 = 2^{-5}$

$\texttt{Offset} = 2, r'_{\text{B}} = 1, r'_{\text{U}} = 1, r'_{\text{M}} = 4, r'_\ell = 0,\ p' = 2^{-6},\ r'[\texttt{x}] = 1,\ q'^2 = 1,\ \text{Cr}'[\texttt{x}] = p'r'q'^2 = 2^{-6}$

$\text{Cr}_t[\texttt{x}] = 2^{-11}$ for all $\texttt{x} \in \mathbb{F}_2^4 \setminus \{0\}$

δ_4 : 00002020000000000000000000000000	δ_5 : 00000440000000000000000000000000
ω_9 : 0000000000000000000000000000x0xx	
δ'_2 : 00000000000044000000000000000000	δ'_4 : 02020000000000000000000000000000
ω'_8 : 0000000000000000000000000000x0xx	

5-Round Distinguisher 1, Figure 31 ✔💻

$\texttt{Offset} = 2, r_{\text{B}} = 1, r_{\text{U}} = 1, r_{\text{M}} = 4, r_\ell = 0,\ p = 2^{-9},\ r[\texttt{x}] = 1,\ q^2 = 1, \text{Cr}[\texttt{x}] = prq^2 = 2^{-9}$

$\texttt{Offset} = 2, r'_{\text{B}} = 1, r'_{\text{U}} = 1, r'_{\text{M}} = 4, r'_\ell = 0,\ p' = 2^{-6},\ r'[\texttt{x}] = 1,\ q'^2 = 1,\ \text{Cr}'[\texttt{x}] = p'r'q'^2 = 2^{-6}$

$\text{Cr}_t[\texttt{x}] = 2^{-15}$ for all $\texttt{x} \in \mathbb{F}_2^4 \setminus \{0\}$

δ_4 : 00000000000000000000066600000000	δ_5 : 00000000000000000000622400000000
ω_9 : 00000000xx0x00000000000000000000	
δ'_2 : 00000000010100000000000000000000	δ'_4 : 00000000000000008008000000000000
ω'_8 : 00000000xx0x00000000000000000000	

5-Round Distinguisher 2, Figure 32

$\texttt{Offset} = 2, r_{\text{B}} = 1, r_{\text{U}} = 1, r_{\text{M}} = 4, r_\ell = 0,\ p = 2^{-9},\ r[\texttt{x}] = 1,\ q^2 = 1, \text{Cr}[\texttt{x}] = prq^2 = 2^{-5}$

$\texttt{Offset} = 2, r'_{\text{B}} = 1, r'_{\text{U}} = 1, r'_{\text{M}} = 4, r'_\ell = 0,\ p' = 2^{-6},\ r'[\texttt{x}] = \frac{\texttt{QDLCT[0x2][x]}}{2^{16}},\ q'^2 = 1,\ \text{Cr}'[\texttt{x}] = p'r'[x]q'^2 = 2^{-5} \cdot r'[\texttt{x}]$

$\text{Cr}_t[\texttt{x}] = 2^{-10} \cdot \frac{\texttt{QDLCT[0x2][x]}}{2^{16}}$ for all $\texttt{x} \in \mathbb{F}_2^4 \setminus \{0\}$

δ_4 : 00000000000000000000002200000000	δ_5 : 00000000000000000000666000000000
ω_9 : 00000000xx0x00000000000000000000	
δ'_2 : 00000000002200000000000000000000	δ'_4 : 00000000000000000000200200000000
ω'_8 : 00000000xx0x00000000000000000000	

5-Round Distinguisher 3, Figure 33

$\texttt{Offset} = 2, r_{\text{B}} = 1, r_{\text{U}} = 1, r_{\text{M}} = 4, r_\ell = 0,\ p = 2^{-4},\ r[\texttt{x}] = \frac{\texttt{QDLCT[0x1][x]}}{2^{16}},\ q^2 = 1, \text{Cr}[\texttt{x}] = prq^2 = 2^{-4} \cdot r[\texttt{x}]$

$\texttt{Offset} = 2, r'_{\text{B}} = 1, r'_{\text{U}} = 1, r'_{\text{M}} = 4, r'_\ell = 0,\ p' = 2^{-6},\ r'[\texttt{x}] = \frac{\texttt{QDLCT[0x6][x]}}{2^{16}},\ q'^2 = 1,\ \text{Cr}'[\texttt{x}] = p'r'[x]q'^2 = 2^{-4} \cdot r'[\texttt{x}]$

$\text{Cr}_t[\texttt{x}] = 2^{-8} \cdot \frac{\texttt{QDLCT[0x1][x]}}{2^{16}} \cdot \frac{\texttt{QDLCT[0x6][x]}}{2^{16}}$ for all $\texttt{x} \in \mathbb{F}_2^4 \setminus \{0\}$

δ_4 : 00000000000000002020000000000000	δ_5 : 00000000444000000000000011010000
ω_9 : 0000000000000000xx0x000000000000	
δ'_2 : 00000000000000000000002200000000	δ'_4 : 00000000660600000000000000000000
ω'_8 : 0000000000000000xx0x000000000000	

Table 13: DL distinguishers for 7 rounds of Orthros-PRF.

7-Round Distinguisher 0, Figure 34

$\texttt{Offset} = 0, r_{\mathrm{B}} = 1, r_{\mathrm{U}} = 3, r_{\mathrm{M}} = 4, r_\ell = 0,\ p = 2^{-18},\ q^2 = 1,\ \mathrm{Cr}[\mathtt{x}] = pr[x]q^2 = 2^{-18} \cdot r[\mathtt{x}]$

$\texttt{Offset} = 0, r'_{\mathrm{B}} = 1, r'_{\mathrm{U}} = 3, r'_{\mathrm{M}} = 4, r'_\ell = 0,\ p' = 2^{-18},\ q'^2 = 1,\ \mathrm{Cr}'[\mathtt{x}] = p'r'[x]q'^2 = 2^{-18} \cdot r'[\mathtt{x}]$

$r[\mathtt{x}] = [2^{-8.02}, 2^{-7.30}, 2^{-7.20}, 2^{-8.19}, 2^{-7.68}, 2^{-7.81}, 2^{-7.80}, 2^{-7.42}, 2^{-8.15}, 2^{-7.70}, 2^{-7.05}, 2^{-7.28}, 2^{-7.77}, 2^{-7.22}, 2^{-7.89}]$

$r'[\mathtt{x}] = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$

$\mathrm{Cr}_t[\mathtt{x}] = [2^{-44.02}, 2^{-43.30}, 2^{-43.20}, 2^{-44.19}, 2^{-43.68}, 2^{-43.81}, 2^{-43.80}, 2^{-43.42}, 2^{-44.15}, 2^{-43.70}, 2^{-43.05}, 2^{-43.28}, 2^{-43.77}, 2^{-43.22}, 2^{-43.89}]$

δ_1 : 00000000000000440000000000000000	δ_4 : 40440000000000000000110100000000
ω_8 : 00000000xx0x0000000000000000000000	
δ'_1 : 00000000220000000000000000000000	δ'_4 : 00000000000000008008000000000000
ω'_8 : 00000000xx0x0000000000000000000000	

7-Round Distinguisher 1, Figure 35

$\texttt{Offset} = 0, r_{\mathrm{B}} = 1, r_{\mathrm{U}} = 3, r_{\mathrm{M}} = 4, r_\ell = 0,\ p = 2^{-27},\ q^2 = 1,\ \mathrm{Cr}[\mathtt{x}] = pr[x]q^2 = 2^{-27} \cdot r[\mathtt{x}]$

$\texttt{Offset} = 0, r'_{\mathrm{B}} = 1, r'_{\mathrm{U}} = 3, r'_{\mathrm{M}} = 4, r'_\ell = 0,\ p' = 2^{-17},\ q'^2 = 1,\ \mathrm{Cr}'[\mathtt{x}] = p'r'[x]q'^2 = 2^{-17} \cdot r'[\mathtt{x}]$

$r[\mathtt{x}] = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$

$r'[\mathtt{x}] = -1 \cdot [2^{-4.02}, 2^{-3.93}, 2^{-4.05}, 2^{-3.81}, 2^{-3.83}, 2^{-3.75}, 2^{-3.85}, 2^{-3.57}, 2^{-4.12}, 2^{-4.17}, 2^{-4.25}, 2^{-3.66}, 2^{-4.25}, 2^{-3.45}, 2^{-4.22}], r'[x] = \frac{\texttt{QDLCT}[0x2][x]}{2^{16}}$

$\widehat{\mathrm{Cr}}_t[\mathtt{x}] = -1 \cdot [2^{-48.02}, 2^{-47.93}, 2^{-48.05}, 2^{-47.81}, 2^{-47.83}, 2^{-47.75}, 2^{-47.85}, 2^{-47.57}, 2^{-48.12}, 2^{-48.17}, 2^{-48.25}, 2^{-47.66}, 2^{-48.25}, 2^{-47.45}, 2^{-48.22}]$

δ_1 : 00000000000000000000066600000000	δ_4 : 00000000000000000000200200000000
ω_8 : 0000x0xx00000000000000000000000	
δ'_1 : 00000000101000000000000000000000	δ'_4 : 00000000000000000000220200000111
ω'_8 : 0000x0xx00000000000000000000000	

7-Round Distinguisher 2, Figure 36

$\texttt{Offset} = 0, r_{\mathrm{B}} = 1, r_{\mathrm{U}} = 3, r_{\mathrm{M}} = 4, r_\ell = 0,\ p = 2^{-17}, r[\mathtt{x}] = 1,\ q^2 = 1,\ \mathrm{Cr}[\mathtt{x}] = pr[x]q^2 = 2^{-17}$

$\texttt{Offset} = 0, r'_{\mathrm{B}} = 1, r'_{\mathrm{U}} = 3, r'_{\mathrm{M}} = 4, r'_\ell = 0,\ p' = 2^{-20},\ q'^2 = 1,\ \mathrm{Cr}'[\mathtt{x}] = p'r'[x]q'^2$

$r'[\mathtt{x}] = -1 \cdot [\approx 0, 2^{-12.58}, 2^{-13.13}, 2^{-13.35}, 2^{-12.46}, 2^{-13.25}, 2^{-13.42}, 2^{-11.75}, 2^{-14.73}, 2^{-13.67}, 2^{-12.93}, 2^{-13.04}, 2^{-11.83}, \approx 0, 2^{-11.71}]$

$\mathrm{Cr}_t[\mathtt{x}] = [\approx 0, 2^{-49.58}, 2^{-50.13}, 2^{-50.35}, 2^{-49.46}, 2^{-50.25}, 2^{-50.42}, 2^{-48.75}, 2^{-51.73}, 2^{-50.67}, 2^{-49.93}, 2^{-50.04}, 2^{-48.83}, \approx 0, 2^{-48.71}]$

δ_1 : 00000404000000000000000000000000	δ_4 : 00000000998100000000000000000000
ω_8 : 0000000000000000xx0x0000000000000	
δ'_1 : 00000000008800000000000000000000	δ'_4 : 88800000440000008008000000000000
ω'_8 : 0000000000000000xx0x0000000000000	

7-Round Distinguisher 3, Figure 37

$\texttt{Offset} = 0, r_{\mathrm{B}} = 1, r_{\mathrm{U}} = 3, r_{\mathrm{M}} = 4, r_\ell = 0,\ p = 2^{-26},\ q^2 = 1,\ \mathrm{Cr}[\mathtt{x}] = pr[x]q^2 = 2^{-26} \cdot r[\mathtt{x}]$

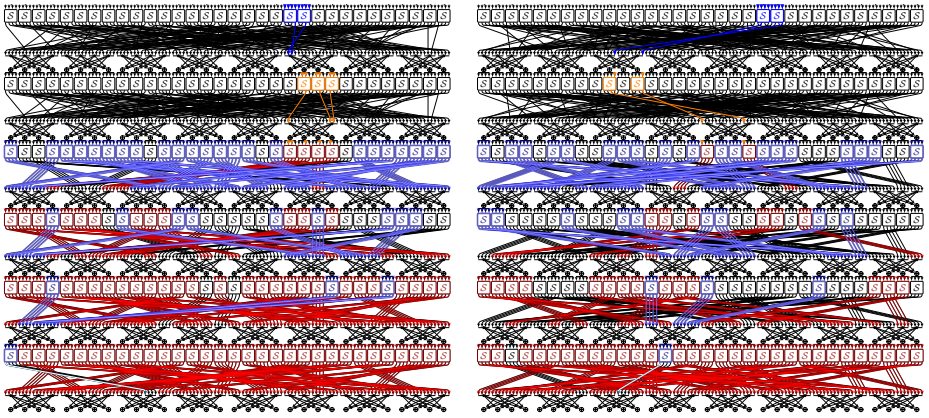
$\texttt{Offset} = 0, r'_{\mathrm{B}} = 1, r'_{\mathrm{U}} = 3, r'_{\mathrm{M}} = 4, r'_\ell = 0,\ p' = 2^{-18},\ q'^2 = 1,\ \mathrm{Cr}'[\mathtt{x}] = p'r'[x]q'^2 = 2^{-18} \cdot r'[\mathtt{x}]$

$\widehat{r}[\mathtt{x}] = -1 \cdot [2^{-4.14}, 2^{-3.83}, 2^{-3.72}, 2^{-4.24}, 2^{-3.92}, 2^{-4.02}, 2^{-3.98}, 2^{-3.81}, 2^{-4.12}, 2^{-3.91}, 2^{-3.61}, 2^{-3.71}, 2^{-3.96}, 2^{-3.72}, 2^{-4.09}], r[\mathtt{x}] = \frac{\texttt{QDLCT}[0x4][x]}{2^{16}}$

$\widehat{r'}[\mathtt{x}] = -1 \cdot [2^{-4.00}, 2^{-3.79}, 2^{-4.09}, 2^{-4.05}, 2^{-4.00}, 2^{-3.98}, 2^{-4.07}, 2^{-3.59}, 2^{-4.39}, 2^{-4.30}, 2^{-3.91}, 2^{-3.79}, 2^{-3.57}, 2^{-3.71}, 2^{-3.64}], r'[x] = \frac{\texttt{QDLCT}[0x8][x]}{2^{16}}$

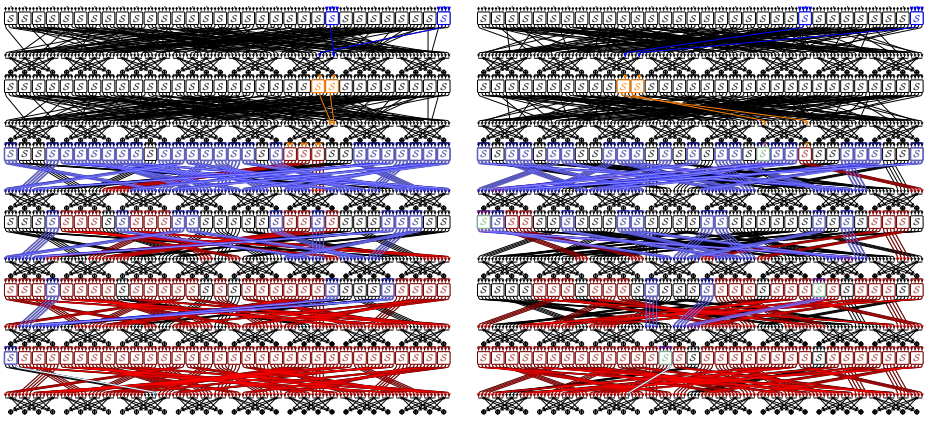
$\mathrm{Cr}_t[\mathtt{x}] = -1 \cdot [2^{-52.14}, 2^{-51.62}, 2^{-51.81}, 2^{-52.29}, 2^{-51.92}, 2^{-52.00}, 2^{-52.05}, 2^{-51.40}, 2^{-52.51}, 2^{-52.21}, 2^{-51.52}, 2^{-51.50}, 2^{-51.53}, 2^{-51.43}, 2^{-51.73}]$

δ_1 : 00002020000000000000000000000000	δ_4 : 30300000004400004cc8000000000000
ω_8 : 0000000000000000000000000xx0x0000	
δ'_1 : 00000000000044000000000000000000	δ'_4 : 00000000200200000000888000000000
ω'_8 : 0000000000000000000000000xx0x0000	



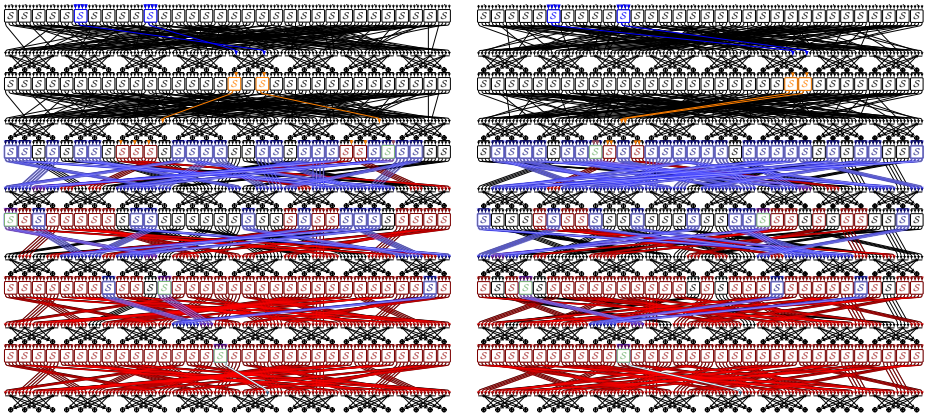
(a) Branch1 of 6-round Orthros (b) Branch2 of 6-round Orthros

Fig. 31: 6-Round DL Attack 1 on Orthros



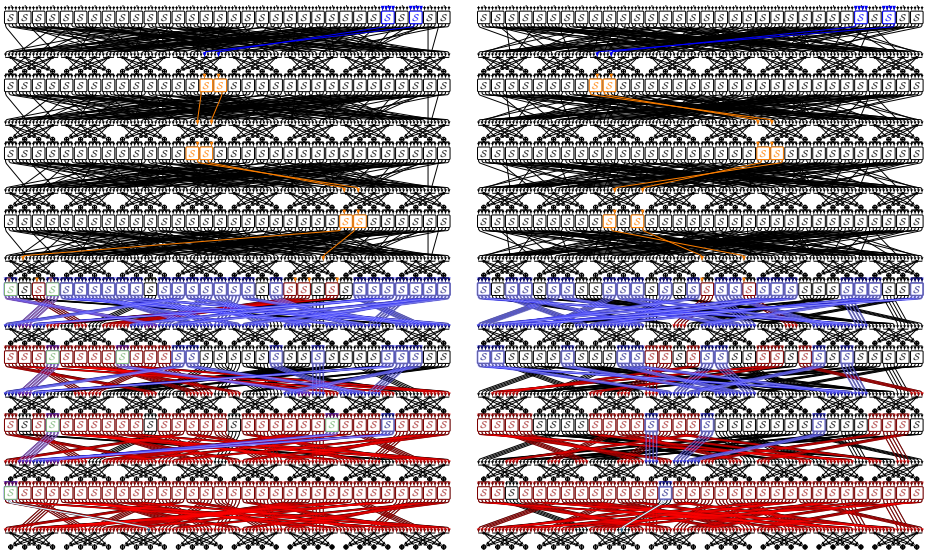
(a) Branch1 of 6-round Orthros (b) Branch2 of 6-round Orthros

Fig. 32: 6-Round DL Attack 2 on Orthros



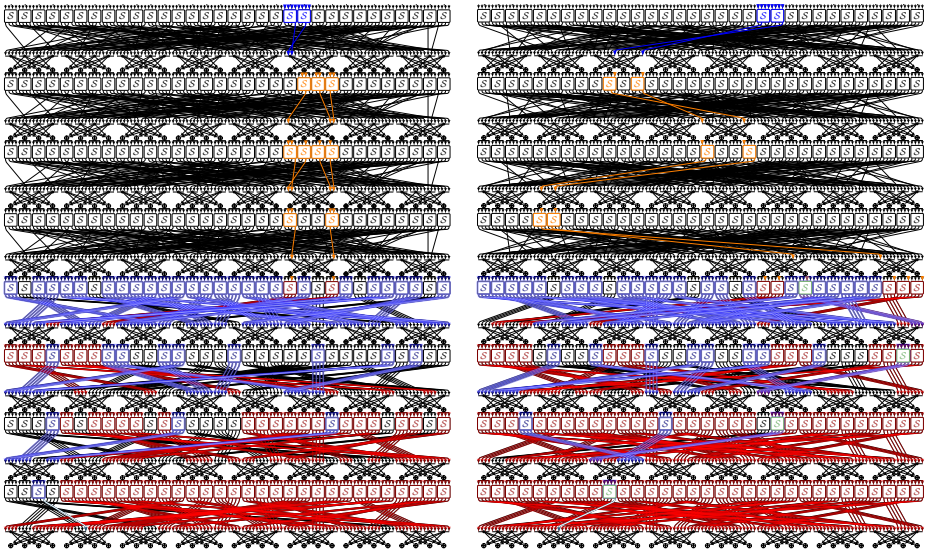
(a) Branch1 of 6-round Orthros (b) Branch2 of 6-round Orthros

Fig. 33: 6-Round DL Attack 3 on Orthros



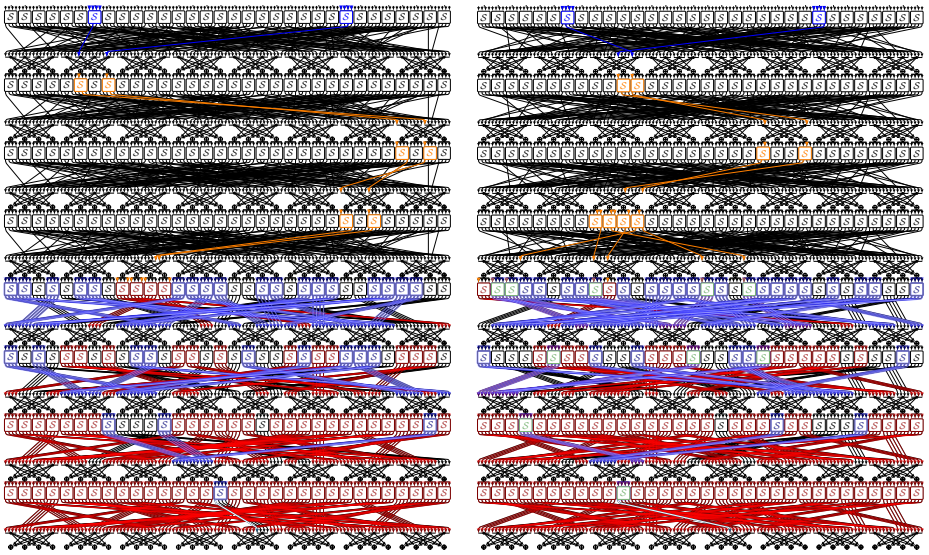
(a) Branch1 of 8-round Orthros (b) Branch2 of 8-round Orthros

Fig. 34: 8-Round DL Attack 0 on Orthros



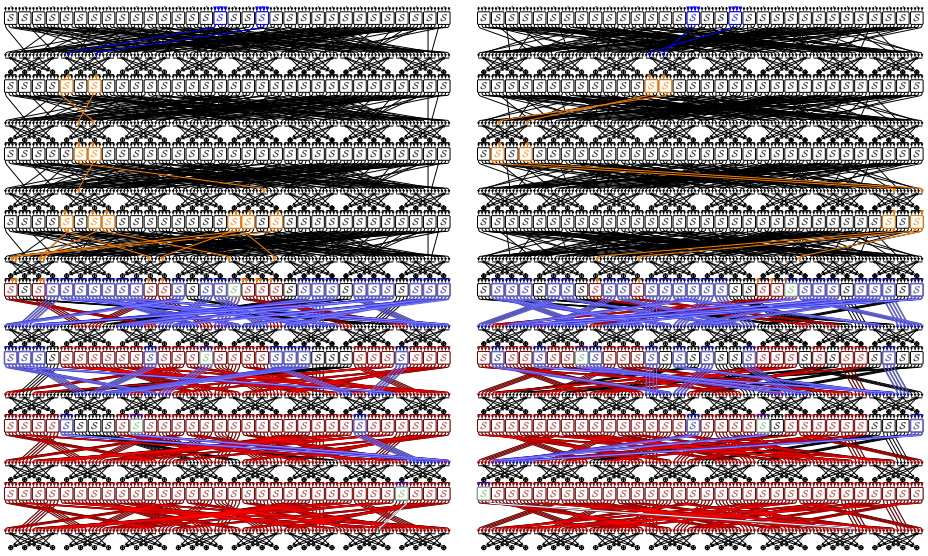
(a) Branch1 of 8-round Orthros (b) Branch2 of 8-round Orthros

Fig. 35: 8-Round DL Attack 1 on Orthros



(a) Branch1 of 8-round Orthros (b) Branch2 of 8-round Orthros

Fig. 36: 8-Round DL Attack 2 on Orthros



(a) Branch1 of 8-round Orthros (b) Branch2 of 8-round Orthros

Fig. 37: 8-Round DL Attack 3 on Orthros