# A Generalized Framework for Conditional Linear Cryptanalysis and Its Application to AES-Like Ciphers

Cheng Che, Tian Tian✉, Jing Yang, Fan Yang

Information Engineering University, Zhengzhou, China
tiantian_d@126.com

**Abstract.** Conditional linear cryptanalysis represents an extension of linear cryptanalysis and has been applied to DES and AES. Notably, it enables the construction of a linear distinguisher for 4-round AES, which is considered unattainable through standard linear cryptanalysis. The underlying principle is that the correlation of a linear approximation can be improved when the data is restricted to a specific subspace or subset, thereby allowing more effective linear cryptanalysis. The critical challenge in conditional linear cryptanalysis lies in identifying appropriate conditions to impose on the data; however, previous methods rely on ad hoc strategies that depend heavily on expert intuition, which limits their generalization and application.

In this paper, we propose a generalized framework for conditional linear cryptanalysis. The core tool is the *conditional linear approximation table* (CLAT), which quantifies linear correlations within constrained data subspaces. We further define a metric termed *conditional linear weight*, which balances the gain in correlation against the overhead of data filtering, thereby offering a quantitative measure of resistance against conditional linear cryptanalysis. Based on the CLAT and conditional linear weight, we develop an MILP-based automatic search model for conditional linear trails and devise systematic approaches for mounting distinguishing and key recovery attacks using these trails. Applying our framework to AES, Rijndael-256, ARIA, LED, Midori-128, and SKINNY-128, we demonstrate that conventional full-space bounds do not guarantee resistance against conditional linear cryptanalysis, and we propose improved linear attacks.

Our framework provides systematic approaches and automatic tools for conditional linear cryptanalysis. It enables cryptanalysts to gain deeper insights into the statistical linear properties of cryptographic primitives and serves as a useful evaluation technique for new designs.

**Keywords:** Conditional Linear Cryptanalysis · CLAT · AES · Rijndael-256 · ARIA · LED · Midori-128 · SKINNY-128

## 1 Introduction

Linear cryptanalysis was introduced by Matsui at EUROCRYPT'93 [27] and has since become a principal technique for analyzing symmetric cryptographic

primitives. It identifies non-trivial nonrandomness of an expression involving plaintext, ciphertext, and key bits, and further exploits such statistical properties to distinguish the cipher from a random permutation or to recover secret key bits. Research in linear cryptanalysis primarily focuses on searching for high-quality linear distinguishers and developing advanced key recovery strategies.

*Search for Linear Distinguishers.* The search for linear distinguishers is the foundation of linear cryptanalysis, as the correlation of a distinguisher directly determines the data complexity and the success probability of the linear attack. Initially, linear distinguishers were primarily discovered through manual analysis of the algebraic and structural properties of cryptographic algorithms [27,21,20]. Consequently, the derived linear approximations often suffered from weak correlations or covered only a limited number of rounds. Over the past decade, automatic search tools such as MILP (Mixed Integer Linear Programming), SAT (Boolean Satisfiability), and CP (Constraint Programming) have been widely employed to search for linear distinguishers [29,33,34]. The main idea is to encode the correlation propagation rules of each underlying component as a set of (in)equalities, which are then solved using off-the-shelf solvers. The feasible solutions correspond to valid linear trails. These automatic search methods do not require detailed knowledge of the global algebraic behavior of a primitive, but rely on accurate modeling of individual components and their interconnections. Consequently, automatic approaches often yield linear approximations that would be challenging to discover through manual analysis.

*Key Recovery Method.* A linear distinguisher can be further used to recover partial key information by appending one additional round after the distinguisher. By guessing the key bits in the last round that are related to the active bits at the output of the distinguisher, one can derive the output values of the distinguisher. Then, the empirical correlation of the distinguisher can be statistically estimated. If the key guess is correct, the empirical correlation will deviate significantly from a uniform distribution; otherwise, it will behave in a random-like manner. Given $N$ plaintext-ciphertext pairs and $l$ guessed key bits, a naive key enumeration approach requires $O(N \cdot 2^l)$ time complexity [27]. To improve efficiency, Matsui (1994) introduced a distillation phase that preprocessed the data only once, thereby reducing the key recovery complexity to $O(N) + O(2^{2l})$ [28]. Collard et al. (2007) further observed that a matrix used to compute empirical correlations had a circulant structure, allowing efficient multiplication of this matrix with a vector using the fast Fourier transform (FFT). This further improved the key recovery complexity to $O(N) + O(l2^l)$ [7]. Subsequent studies have focused on enhancing the FFT-based approach, including extending it to multiple rounds [14], integrating Walsh pruning techniques [13,15], etc.

*Conditional Linear Cryptanalysis.* The effectiveness of linear cryptanalysis is fundamentally determined by the quality of linear approximations of nonlinear operations, making the design of nonlinear components in a cipher particularly critical. These components should achieve a high degree of nonlinearity and

enable the cipher to reach a certain security level after a sufficient number of rounds through proper arrangement and composition. In the design of a symmetric cipher, designers typically estimate the required number of rounds and establish the security bound against linear cryptanalysis by evaluating the so-called linear weights of all trails. These linear weights are computed over the entire input space. However, when the input space of a nonlinear component is restricted to a subspace or subset, the correlation of a linear approximation of the component can become significantly higher. This provides cryptanalysts with additional potential and may break the security bounds originally assumed by designers based on full-space data. Such an approach is known as conditional linear cryptanalysis.

As described above, the rationale of conditional linear cryptanalysis lies in the fact that when the data is restricted to a specific subspace or subset, the correlation of a nonlinear component can be improved. Suppose that the correlation is increased from $c$ to $c^*$ and only the fraction $\alpha$ of the data is used, then the data complexity changes from $\frac{1}{c^2}$ to $\frac{1}{\alpha(c^*)^2}$. If $c^*$ is significantly higher than $c$, the attack complexity can be greatly reduced. For example, Biham and Carmeli in [5] partially encrypted the plaintext by guessing subkey bits and partitioned certain bits of the output to amplify the correlation. Biham and Perle used a similar idea in [6] by conditioning on some special relations of plaintext and ciphertext in Feistel ciphers like DES. Ashur and Takke in [1] further extended the idea to SPN-based ciphers and applied it to AES by only using data within a specific subspace of an S-box. With this method, they gave a 4-round linear distinguisher of AES, which was previously considered unachievable. However, the methods for data filtering discussed in previous work are ad hoc and do not provide clarity on whether better conditional filtering methods exist. In this paper, we delve into this problem and propose a generalized framework for conditional linear cryptanalysis.

### 1.1 Our Contributions

Ashur and Takke identified strong conditional linear correlations for the AES S-box by deeply analyzing the multiplicative inverse function over the finite field $\mathbb{F}_{2^8}$ (Journal of Cryptology, 2025) [1]. In this paper, we follow their line of research and present a generalized framework for conditional linear cryptanalysis, enabling its extension to a wide range of symmetric-key primitives, including AES. Our main contributions are outlined below.

**Generalized Framework for Conditional Linear Cryptanalysis.** We propose a generalized framework for conditional linear cryptanalysis that includes: (i) the conditional linear approximation table (CLAT), a newly introduced tool for characterizing the linear behavior of nonlinear components under specified conditions; (ii) an MILP-based automatic search model designed to efficiently search for high-quality conditional linear trails; and (iii) systematic methodologies for mounting distinguishing and key recovery attacks using the identified conditional linear trails. This framework addresses theoretical gaps in prior

work by introducing the CLAT, enabling refined security bounds and deriving improved attacks on AES-like ciphers.

- *The Conditional Linear Approximation Table (CLAT).* To characterize the influence of conditions on the correlation, we introduce the CLAT, which can be regarded as a condition-aware extension of the linear approximation table (LAT). For a vectorial Boolean function $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$, given a condition set $\mathcal{C}$, the table $CLAT_{\mathcal{C}}$ is a matrix of size $2^n \times 2^m$, where the entry at $(\lambda_i, \lambda_o)$ represents the conditional linear correlation $\mathbb{C}_{\mathcal{C}}(\lambda_i, \lambda_o)$ of the linear approximation $\lambda_i x \oplus \lambda_o S(x)$ under the data subspace filtered by $\mathcal{C}$, with $\lambda_i$ and $\lambda_o$ being the input and output linear masks, respectively. Unlike the LAT, which measures correlations over the entire data space, the CLAT partitions the data based on conditions and stores the correlation over the data within a certain subspace. Therefore, it can reveal statistical properties hidden within subspaces that are not apparent in the full-space LAT.
- *The Conditional Linear Weight.* Based on the CLAT, we introduce a new metric called the conditional linear weight to quantify the data complexity of distinguishing attacks using conditional linear correlations. Given a condition set $\mathcal{C}$, an input linear mask $\lambda_i$, and an output linear mask $\lambda_o$, the conditional linear weight is defined as

$$w_{\mathcal{C}}(\lambda_i, \lambda_o) := -\log_2 \mathbb{C}_{\mathcal{C}}^2(\lambda_i, \lambda_o) + |\mathcal{C}|.$$

  Taking the S-box of AES as an example, the optimal conditional linear weight is 3.83, which is significantly lower than its linear weight of 6. This indicates that its behavior under conditions is less resistant than suggested by the ordinary LAT. This metric provides a useful measure for evaluating the resistance of cryptographic primitives to conditional linear cryptanalysis.
- *The Conditional Linear Trail and Automatic Searching.* We propose a framework for constructing conditional linear trails by decomposing a target cipher $E$ into three sub-ciphers: $E = E_1 \circ E_m \circ E_0$. The sub-ciphers $E_0$ and $E_1$ are analyzed using CLATs to derive conditional linear trails, while the middle part $E_m$ is examined using standard linear trails. We then introduce a new MILP model for the automatic search of conditional linear trails. The route of the best conditional linear trail may differ from that of the best standard linear trail, indicating that the new conditional linear cryptanalysis framework is not a simple extension of traditional linear cryptanalysis.
- *Distinguishing Attacks and Key Recovery Attacks.* Based on the derived conditional linear trails, we present procedures for performing distinguishing attacks and key recovery attacks. For key recovery attacks, we provide two different scenarios according to the number of active S-boxes in $E_0$ and $E_1$. This enables us to flexibly balance the trade-off between attack complexity and the amount of recovered key bit information according to the conditional linear trails of different ciphers. Additionally, we incorporate the fast Walsh-Hadamard transform (FWHT) to accelerate the key recovery process.

**Table 1.** Bounds on the weights of (conditional) linear trails and exact weights of identified trails for some cryptographic primitives.

| Cipher | Type† | | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | Ref. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AES | L | B | 150 | 156 | 180 | 204 | 300 | 306 | 330 | - | - | - | - | - | [29] |
| | | E | 150 | - | - | - | - | - | - | - | - | - | - | - | [34] |
| | CL | B | 106.6 | 151.66 | 169.15 | 186.64 | 256.6 | 301.66 | 319.15 | - | - | - | - | - | Our |
| | | E | **106.6** | - | - | - | - | - | - | - | - | - | - | - | Our |
| Rijndael -256 | L | B | 150 | 246 | 300 | 348 | 390 | 444 | 510 | - | - | - | - | - | [31] |
| | | E | 150 | 246 | - | - | - | - | - | - | - | - | - | - | [34] |
| | CL | B | 106.6 | 176.56 | 267.45 | 313.28 | 374.81 | 415.28 | 477.45 | - | - | - | - | - | Our |
| | | E | 106.6 | 176.56 | - | - | - | - | - | - | - | - | - | - | Our |
| ARIA | L | B | 96 | 102 | 144 | 150 | 192 | 198 | 240 | 246 | 288 | - | - | - | [23] |
| | | E | 96 | 105.2 | 144 | 154.8 | - | - | - | - | - | - | - | - | [34] |
| | CL | B | 78.64 | 97.66 | 126.64 | 145.66 | 174.64 | 193.66 | 224.64 | 241.66 | 270.64 | - | - | - | Our |
| | | E | 78.64 | 100.86 | **126.64** | 150.46 | - | - | - | - | - | - | - | - | Our |
| LED | L | B | 50 | 52 | 60 | 68 | 100 | 102 | 110 | 118 | 150 | - | - | - | [18] |
| | | E | 50 | 58 | 64 | 74 | 100 | 114 | 122 | 134 | 154 | - | - | - | [34] |
| | CL | B | 30 | 50 | 55 | 60 | 80 | 100 | 105 | 110 | 130 | - | - | - | Our |
| | | E | 30 | 50 | **61** | 68 | 86 | 110 | - | - | - | - | - | - | Our |
| Midori -128 | L | B | 32 | 46 | 60 | 70 | 76 | 82 | 100 | 114 | 124 | 134 | 144 | - | [2] |
| | | E | 32 | 46 | 64 | 74 | 84 | 92 | 110 | 124 | 136 | - | - | - | [34] |
| | CL | B | 20 | 32 | 48 | 62 | 70 | 76 | 88 | 100 | 116 | 124 | 136 | - | Our |
| | | E | 20 | 32 | 48 | 68 | 74 | 90 | 102 | 116 | **124** | - | - | - | Our |
| SKINNY -128* | L | B | 16 | 26 | 38 | 50 | 64 | 76 | 86 | 96 | 104 | 110 | 116 | 128 | [4] |
| | | E | 16 | 26 | 38 | 50 | 64 | 78 | 90 | 102 | 112 | 122 | - | - | [34] |
| | CL | B | 10 | 18 | 27 | 38 | 51 | 66 | 77 | 88 | 96 | 105 | 110 | 119 | Our |
| | | E | 10 | 18 | 27 | 38 | 51 | 66 | 85.49 | 96.49 | 105.66 | 118.32 | **125.32** | - | Our |

† L: Linear trails CL: Conditional linear trails
B: Lower bounds E: The exact weights of identified (conditional) linear trails
* In the single-tweakey setting

**Applications to AES-Like Ciphers.** We apply our framework to several AES-like ciphers, including AES, Rijndael-256, ARIA, LED, Midori-128, and SKINNY-128, providing their theoretical security bounds against conditional linear cryptanalysis and developing improved linear attacks. All experimental results and source code are available at https://github.com/CLAT-tool/codes-and-results.

- *New Security Bounds Against Linear Cryptanalysis.* Establishing security bounds against linear cryptanalysis is a crucial task in evaluating the security of symmetric-key primitives. We discover improved linear trails for all the considered ciphers and surpass the security bounds established by standard linear cryptanalysis for AES, ARIA, LED, Midori-128, and SKINNY-128. This indicates that the conventional full-space linear trail bounds do not directly imply security against conditional linear distinguishers. Table 1 summarizes the comparison of our results with previous work, with boldface highlighting the rounds where valid conditional linear approximations surpass previously known valid standard linear approximations.

**Table 2.** Comparison of our results with existing ones regarding distinguishers.

| Cipher | Round | Type | Complexity | Success Probability | Ref. |
|---|---|---|---|---|---|
| AES* | 3 | Linear | $2^{57}$ | 99.8% | [34] |
| | 4 | Conditional Linear | $2^{126.04}$ | 99.8% | [1] |
| | **4** | **Conditional Linear** | $\mathbf{2^{115.60}}$ | 99.8% | **Our** |
| Rijndael-256* | 5 | Linear | $2^{249}$ | 99.8% | [34] |
| | **5** | **Conditional Linear** | $\mathbf{2^{186.28}}$ | 99.8% | **Our** |
| ARIA | 4 | Integral | $2^{96}$ | 75%‡ | [26] |
| | 5 | Differential | $2^{112}$ | 81.6%‡ | [34] |
| | 5 | Linear | $2^{108.2}$ | 99.8% | [34] |
| | **5** | **Conditional Linear** | $\mathbf{2^{106.18}}$ | 99.8% | **Our** |
| LED | 5 | Differential | $2^{59}$‡ | 80.1%‡ | [34] |
| | 5 | Linear | $2^{61}$ | 99.8% | [34] |
| | **5** | **Conditional Linear** | $\mathbf{2^{56.50}}$ | 99.8% | **Our** |
| Midori-128 | 6 | Integral | $2^{72}$ | 75%‡ | [25] |
| | 10 | Differential | $2^{114}$ | 81.6%‡ | [34] |
| | 11 | Linear | $2^{127}$ | 99.8% | [34] |
| | **11** | **Conditional Linear** | $\mathbf{2^{121.83}}$ | 99.8% | **Our** |
| SKINNY-128 | **13** | **Conditional Linear** | $\mathbf{2^{123.30}}$¶ | 99.8% | **Our** |
| | 13 | Boomerang | $2^{112.53}$ | 81.6%‡ | [11] |
| | 14 | Differential† | $2^{120}$ | 81.4%‡ | [34] |

* We only list the linear distinguishers for AES and Rijndael-256.
† The clustering effect is considered.
‡ The success probability is calculated as $\frac{p_0+p_1}{2}$, where $p_0$ is the probability of correctly classifying the target cipher, and $p_1$ is the probability of correctly classifying a random permutation.
¶ This complexity is effective for at least half of the keys.

- *Distinguishing Attacks.* Based on the discovered conditional linear trails, we present the best known linear distinguishers for these ciphers. Specifically, for ARIA, LED, and Midori-128, these distinguishers are the best known. In Table 2, we compare the newly discovered conditional linear distinguishers with those previously known.
- *Key Recovery Attacks on AES, Rijndael-256, and Midori-128.* We propose key recovery attacks on round-reduced versions of AES, Rijndael-256, and Midori-128. Specifically, we present a 4-round key recovery attack on AES with a complexity of $2^{115.28}$, improving upon the complexity reported in [1] by a factor of $2^{10.44}$. For the first time, we propose linear key recovery attacks against 7-round Rijndael-256 and 12-round Midori-128, with complexities of $2^{241.09}$ and $2^{115.48}$, respectively. The results are summarized in Table 3.

**Table 3.** Linear key recovery attacks on AES, Rijndael-256 and Midori-128.

| Cipher | Round | Time | Data | Space | Reference |
|---|---|---|---|---|---|
| AES | 4 | $2^{125.72}$<br>$2^{115.28}$ | $2^{125.72}$<br>$2^{115.28}$ | $2^{33}$<br>$2^{61}$ | [1]<br>Our |
| Rijndael-256 | 7 | $2^{241.12}$ | $2^{241.06}$ | $2^{228}$ | Our |
| Midori-128 | 12 | $2^{115.48}$ | $2^{115.47}$ | $2^{100}$ | Our |

### 1.2 Organization of the Paper

The remainder of the paper is organized as follows. In Section 2, we first present the necessary preliminaries. Then, in Section 3, we introduce the conditional linear approximation table, along with several propositions, and present the concept of conditional linear weights. Section 4 presents the construction and search methods for conditional linear trails, as well as distinguishing attacks and key recovery methods based on these trails. In Section 5, we derive conditional linear bounds for several AES-like ciphers and identify valid conditional linear distinguishers. We then apply the conditional linear cryptanalysis method to AES, Rijndael-256, and Midori-128, presenting specific key recovery attacks. Section 6 discusses our new framework and raises several open questions for future research. Finally, Section 7 concludes the paper.

## 2 Preliminaries

### 2.1 Notations

Let $\mathbb{F}_2$ denote the field with two elements as $\mathbb{F}_2 = \{0, 1\}$, and $\mathbb{F}_2^n$ as the $n$-dimensional binary vector space. For $a, b \in \mathbb{F}_2^n$, their concatenation is denoted as $a||b$ and their dot product is defined as $\sum_{i=0}^{n-1} a[i] \cdot b[i]$, where $a[i]$ (resp. $b[i]$) represents the $i$-th bit of $a$ (resp. $b$). The notation $a[0, \ldots, r-1]$ denotes the truncated vector consisting of the first $r$ elements of $a$. For the symbol $|A|$, if $A$ is a set, $|A|$ denotes the number of elements in $A$; if $A$ is a real number, $|A|$ denotes its absolute value. For the product of two numbers, we use "$\cdot$" or "$\times$" to denote multiplication. Given a set $\mathcal{S} \subset \mathbb{F}_2^n$ and a Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$, the correlation of $f$ over the sample space $\mathcal{S}$ is defined as

$$\mathtt{Cor}_{\mathcal{S}}(f) := \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} (-1)^{f(x)} = \Pr(f(x) = 0 | x \in \mathcal{S}) - \Pr(f(x) = 1 | x \in \mathcal{S}).$$

### 2.2 Linear Cryptanalysis

Linear cryptanalysis exploits linear relations between plaintext and ciphertext and such linear relations are usually called linear approximations.

**Definition 1 (Linear Approximation [27]).** *For a vectorial Boolean function $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$, a linear approximation of $S$ is defined by a pair of masks $(\lambda_i, \lambda_o)$, i.e., $\lambda_i \cdot x \oplus \lambda_o \cdot S(x)$, where $\lambda_i \in \mathbb{F}_2^n$ and $\lambda_o \in \mathbb{F}_2^m$. The correlation of the linear approximation is computed as*

$$\mathbb{C}(\lambda_i, \lambda_o) := \mathtt{Cor}(\lambda_i \cdot x \oplus \lambda_o \cdot S(x)) = \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} (-1)^{\lambda_i \cdot x \oplus \lambda_o \cdot S(x)}.$$

The correlation of a linear approximation quantifies the statistical linear properties between the input and output of nonlinear components (e.g., S-boxes). The values of correlations range from $-1$ to 1, and higher absolute values indicate stronger linear relations between the masked input and output.

**Definition 2 (Linear Approximation Table [27]).** *For a vectorial Boolean function $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$, the LAT of $S$ is a $2^n \times 2^m$ table of which rows and columns are indexed by the input mask $\lambda_i \in \mathbb{F}_2^n$ and output mask $\lambda_o \in \mathbb{F}_2^m$ of $S$, respectively. The entry at position $(\lambda_i, \lambda_o)$ is defined as*

$$LAT(\lambda_i, \lambda_o) := |LAT_0(\lambda_i, \lambda_o)| - |LAT_1(\lambda_i, \lambda_o)|,$$

*where $LAT_b(\lambda_i, \lambda_o) = \{x \in \mathbb{F}_2^n : \lambda_i \cdot x \oplus \lambda_o \cdot S(x) = b\}$.*

The LAT provides an efficient method to derive the correlation of a linear approximation for a single S-box, i.e., $\mathbb{C}(\lambda_i, \lambda_o) = LAT(\lambda_i, \lambda_o)/2^n$. Its primary purpose is to precompute the correlations of linear approximations corresponding to all possible input and output linear masks, thereby providing a foundation for constructing round-wise linear approximations. For block ciphers, multi-round linear approximations are constructed by combining single-round linear approximations, with the overall correlation calculated using the piling-up lemma [27].

**Definition 3 (Linear Trail [8]).** *Let an iterative key-alternating block cipher $E_K : \{0,1\}^n \to \{0,1\}^n$ be composed of $r$ round functions $R_0, R_1, \ldots, R_{r-1}$ and each round is XORed with subkeys $K_0, K_1, \ldots, K_{r-1}$, respectively. Then, the $r$-round linear trail of $E_K$ is a sequence*

$$\Lambda = (\lambda_0 \xrightarrow{R_0} \lambda_1 \xrightarrow{R_1} \cdots \xrightarrow{R_{r-1}} \lambda_r) := (\lambda_0, \lambda_1, \cdots, \lambda_r),$$

*where $\lambda_0 \in \{0,1\}^n$ and $\lambda_r \in \{0,1\}^n$ are linear masks for the plaintext and ciphertext, respectively, and $\lambda_i, 1 \le i \le r-1$, is the input linear mask for the $i$-th round. The correlation of the linear trail $\Lambda$ is defined*

$$\mathbb{C}(\Lambda) = \prod_{i=1}^{r} \mathbb{C}(\lambda_{i-1}, \lambda_i),$$

*where $\mathbb{C}(\lambda_{i-1}, \lambda_i)$ is the correlation of the linear approximation of the $i$-th round.*

For a fixed external mask $(\lambda_0, \lambda_r)$, there may exist many linear trails with different intermediate masks. The collection of these linear trails is called the

*linear hull*, and the correlation of the linear approximation $(\lambda_0, \lambda_r)$ is obtained by summing the signed correlations of all trails in the hull:

$$\mathbb{C}(\lambda_0, \lambda_r) = \sum_{\lambda_1, \ldots, \lambda_{r-1}} (-1)^{S_\Lambda} \mathbb{C}(\lambda_0, \lambda_1, \ldots, \lambda_{r-1}, \lambda_r),$$

where the signs $S_\Lambda \in \{0, 1\}$ depend on the round keys. The *linear hull effect* refers to the fact that the total correlation of a linear approximation is generally not determined by a single trail, but by the signed sum of all trails sharing the same external masks. Since the signs of the trail contributions depend on the round keys, the total correlation is key-dependent: for different keys, the trail contributions may either cancel each other or reinforce each other.

Therefore, a single dominant trail can approximate the total correlation only when the remaining trails are negligible. Nevertheless, as shown in [24], under the usual heuristic assumption that the signs of trail contributions are independent and uniformly distributed with respect to the key, if a single trail has correlation $\mathbb{C}$, then for at least half of the keys, the complexity of the corresponding linear attack is bounded by approximately $c/\mathbb{C}^2$, where $c$ is a small constant.

**Definition 4 (Linear Weight).** *The linear weight $w$ of a linear approximation is defined as the negative binary logarithm of its squared linear correlation, that is, $w := -\log_2 \mathbb{C}^2$. The optimal linear weight of an S-box $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$ is defined as*

$$\mathcal{L}(S) := \min_{(\lambda_i, \lambda_o) \in \mathbb{F}_2^{n+m} \setminus \{\mathbf{0}\}} \{-\log_2 \mathbb{C}^2(\lambda_i, \lambda_o)\}.$$

Using the multiplication of the optimal linear weight of one S-box by the number of active S-boxes can provide a lower bound on the optimal overall linear weight. The linear weight can be used to give a rough evaluation of the resistance of a cipher against linear cryptanalysis. For a linear trail with weight $w$, the data complexity of a distinguishing attack is $\mathcal{O}(2^w)$.

### 2.3 Automatic Tool for Searching Linear Trails

In recent years, automatic search tools such as MILP have become the main approach for searching linear trails [29,33,17]. The core idea is to represent the rules of linear propagation as a set of mathematical (in)equalities and constraints, and then employ off-the-shelf solvers to find feasible solutions. These feasible solutions correspond to valid linear trails of the cipher.

One critical focus in automatic searching is the modeling of nonlinear operations such as the S-box. For a bijective S-box from $\mathbb{F}_2^n$ to $\mathbb{F}_2^n$, suppose that the input and output linear masks are $(x_0, x_1, \ldots, x_{n-1})$ and $(y_0, y_1, \ldots, y_{n-1})$, respectively. If the values of these linear masks are not all zero, such an S-box is referred to as active. In most cases, the linear weight of a linear trail is positively correlated with the number of active S-boxes, and more active S-boxes generally indicate smaller overall correlation. Therefore, one common objective of the MILP modeling is to minimize the number of active S-boxes. This can be

achieved by assigning a flag variable $A_t$ for each S-box and using 1/0 to indicate its activeness/inactiveness. The value of $A_t$ is determined through the clause that $A_t = 1$ if and only if the input linear mask $(x_0, x_1, \ldots, x_{n-1})$ is a nonzero vector. Therefore, one widely used objective function in the MILP modeling is minimizing $\sum A_t$.

Based on the trails of active S-boxes, one can further search for bitwise linear trails using refined constraints. Alternatively, one may directly search for the best bitwise linear trail without relying on active S-box trails; however, this approach often results in a significantly larger search space, making it impractical. For more details on constructing (in)equalities for different cipher operations to describe linear propagation, please refer to [29,33,17].

## 3 The Conditional Linear Approximation Table

In this section, we introduce the conditional linear approximation table (CLAT) and the conditional linear weight, and discuss their properties.

### 3.1 The Conditional Linear Approximation Table and Its Properties

Conditional linear cryptanalysis [6,1] of DES and AES shows that when the input data of a linear approximation is filtered through certain conditions, the statistical correlation can be improved. In this paper, we use a generalization of the approach in [1] to impose conditions on the data such that the selected data form a certain subspace. Specifically, each condition is represented as a pair $(\mathcal{T}, t)$, where

- $\mathcal{T}$ is a binary vector whose dimension matches that of the data to be filtered and is called the *condition mask*;
- $t \in \mathbb{F}_2$ is a binary value and is called the *condition value*.

The data to be filtered can be the input, the output, or the concatenation of the input and output of a function. The constraint of a single condition $(\mathcal{T}, t)$ on the data $x$ is expressed as $\mathcal{T} \cdot x = t$. For a given value of $x$, if $\mathcal{T} \cdot x$ equals the condition value $t$, then $x$ is retained; otherwise, it is discarded. Therefore, if $x$ is uniformly distributed, only half of the data will be selected to perform further cryptanalysis under one condition.

In practical applications, one can impose a set of conditions

$$\mathcal{C} = \{(\mathcal{T}_0, t_0), (\mathcal{T}_1, t_1), \ldots, (\mathcal{T}_{c-1}, t_{c-1})\}$$

to achieve stricter data filtering, where the condition masks $\mathcal{T}_0, \mathcal{T}_1, \ldots, \mathcal{T}_{c-1}$ are linearly independent over $\mathbb{F}_2$. Only those values of $x$ that satisfy all conditions in the set, i.e.,

$$\mathcal{T}_j \cdot x = t_j, \quad j \in \{0, 1, \ldots, c-1\}$$

are retained. Although this discards a larger fraction of the data, the correlation over the resulting conditioned subset may be improved. When a linear approximation is evaluated on a data subset satisfying specific conditions, it is called

a *conditional linear approximation*, and the corresponding correlation restricted to this subset is referred to as the *conditional linear correlation*.

To characterize the influence of these conditions on the correlation, we introduce the conditional linear approximation table (CLAT). The CLAT extends the LAT by imposing different conditions on the data and deriving updated correlations. It enables cryptanalysts to characterize the linear correlations after filtering data, providing a foundational tool for conditional linear cryptanalysis.

**Definition 5 (Conditional Linear Approximation Table).** *Given a vectorial Boolean function $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$, the conditional linear approximation table $CLAT_{\mathcal{C}}$ of $S$ under the condition set $\mathcal{C} = \{(\mathcal{T}_0, t_0), (\mathcal{T}_1, t_1), \ldots, (\mathcal{T}_{c-1}, t_{c-1})\}$ where $\mathcal{T}_i \in \mathbb{F}_2^{n+m}, t_i \in \mathbb{F}_2$, is a $2^n \times 2^m$ table of which the rows and columns are indexed by the input mask $\lambda_i$ and output mask $\lambda_o$ of $S$, respectively. The entry at position $(\lambda_i, \lambda_o)$ is defined as*

$$CLAT_{\mathcal{C}}(\lambda_i, \lambda_o) := |CLAT_{\mathcal{C},0}(\lambda_i, \lambda_o)| - |CLAT_{\mathcal{C},1}(\lambda_i, \lambda_o)|,$$

*where $CLAT_{\mathcal{C},b}(\lambda_i, \lambda_o) = \{x \in \mathbb{F}_2^n : \lambda_i \cdot x \oplus \lambda_o \cdot S(x) = b$ and $\mathcal{T}_j \cdot (x||S(x)) = t_j, j = 0, ..., c-1\}$.*

Then based on Definition 5, the conditional correlation under $\mathcal{C}$ is

$$\mathbb{C}_{\mathcal{C}}(\lambda_i, \lambda_o) = CLAT_{\mathcal{C}}(\lambda_i, \lambda_o)/N_{\mathcal{C}},$$

where $N_{\mathcal{C}}$ denotes the number of $x$ values that satisfy the condition set $\mathcal{C}$.

*Complexity.* For the vectorial Boolean function $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$, given the condition mask set $\{\mathcal{T}_0, \mathcal{T}_1, \ldots, \mathcal{T}_{c-1}\}$, there are $2^c$ CLATs, as there are $2^c$ different condition values. We exhaust all inputs, as well as all input and output masks, to compute these CLATs. For each input, we compute the condition value and store the results in the CLAT corresponding to that condition value. Therefore, the total complexity for computing these $2^c$ CLATs is $\mathcal{O}(2^{2n+m})$. We note that the constant involved in the $\mathcal{O}(\cdot)$ notation is negligible since the dot products of vectors can be efficiently derived using a small precomputed lookup table.

**Properties of the CLAT.** Next, we present several basic properties of the CLAT that are useful for constructing and analyzing conditional linear trails.

**Proposition 1.** *If the condition set $\mathcal{C}$ is empty, then the CLAT is reduced to the LAT.*

**Proposition 2.** *Given a condition mask set $\{\mathcal{T}_0, \mathcal{T}_1, \ldots, \mathcal{T}_{c-1}\}$, there are $2^c$ CLATs as each $t_i$ can take either 0 or 1. The sum of the same entries in the $2^c$ $CLAT_{\mathcal{C}}$ equals the value at the same entry in the LAT, i.e.,*

$$LAT(\lambda_i, \lambda_o) = \sum_{(t_0,\ldots,t_{c-1}) \in \mathbb{F}_2^c} CLAT_{\{(\mathcal{T}_0,t_0),\ldots,(\mathcal{T}_{c-1},t_{c-1})\}}(\lambda_i, \lambda_o).$$

**Definition 6 (Equivalence of Linear Approximations).** *Given two linear approximations* $(u_0, v_0)$ *and* $(u_1, v_1)$ *of a vectorial Boolean function* $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$*, if for any* $x \in \mathbb{F}_2^n$ *it holds that*

$$u_0 \cdot x \oplus v_0 \cdot S(x) \oplus u_1 \cdot x \oplus v_1 \cdot S(x) = c,$$

*where* $c \in \mathbb{F}_2$ *is a constant, then we say that the two linear approximations* $(u_0, v_0)$ *and* $(u_1, v_1)$ *are equivalent.*

Straightforwardly, two equivalent linear approximations share the same absolute conditional correlation. Since equivalent linear approximations reveal identical statistical properties, only one is needed for linear cryptanalysis.

**Proposition 3.** *For a vectorial Boolean function* $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$*, given a condition set* $\mathcal{C}$ *with cardinality* $|\mathcal{C}| = c$*, all conditional linear approximations are partitioned into several equivalence classes, each containing* $2^c$ *conditional linear approximations. Particularly, if all condition masks apply only to the input of* $S$*, then for a given output mask* $\lambda_o$*, there exist* $2^c$ *input linear masks that yield equivalent linear approximations.*

*Proof.* Suppose that the condition set $\mathcal{C}$ is denoted as $\{(\mathcal{T}_0, t_0), \ldots, (\mathcal{T}_{c-1}, t_{c-1})\}$. Then for an element $x \in \mathbb{F}_2^n$ satisfying the condition set, it holds that

$$\begin{aligned}
&\lambda_i \cdot x \oplus \lambda_o \cdot S(x) \\
=& \bigoplus_{i=0}^{c-1} a_i(\mathcal{T}_i \cdot (x||S(x)) \oplus t_i) \oplus \lambda_i \cdot x \oplus \lambda_o \cdot S(x) \\
=& \left(\bigoplus_{i=0}^{c-1} a_i \mathcal{T}_i[0, \ldots, n-1] \oplus \lambda_i\right) \cdot x \\
&\qquad \oplus \left(\bigoplus_{i=0}^{c-1} a_i \mathcal{T}_i[n, \ldots, n+m-1] \oplus \lambda_o\right) \cdot S(x) \oplus \bigoplus_{i=0}^{c-1} a_i t_i,
\end{aligned}$$

for all $a_0, a_1, \ldots, a_{c-1} \in \mathbb{F}_2$. This implies that the linear approximations $(\lambda_i, \lambda_o)$ and

$$(\bigoplus_{i=0}^{c-1} a_i \mathcal{T}_i[0, \ldots, n-1] \oplus \lambda_i, \bigoplus_{i=0}^{c-1} a_i \mathcal{T}_i[n, \ldots, n+m-1] \oplus \lambda_o)$$

are equivalent. Since there are $2^c$ possibilities for $(a_0, a_1, \ldots, a_{c-1})$, there are $2^c$ equivalent conditional linear approximations given a condition set of size $c$. □

*Remark 1.* In Definition 5, the condition masks are defined to be applied to both the input and output of the vectorial Boolean function $S$. However, for convenience, we consider the condition masks for the input and output separately in the following cryptanalysis.

**Conditional Linear Weight.** In conditional linear cryptanalysis, an effective attack requires balancing the gain from conditional linear correlation against the data loss caused by filtering. On one hand, the conditional linear correlation directly determines the statistical non-uniformity of the linear trail; a higher correlation implies that fewer data is needed for the attack. The inverse of the squared correlation, $\mathbb{C}_{\mathcal{C}}^{-2}$, is often used as a measure of data complexity. On the other hand, the size $c$ of the condition set $\mathcal{C}$ directly affects the amount of effective data. In a random scenario, the probability that data satisfies $c$ independent conditions is $2^{-c}$. To quantify this relationship and establish a unified conditioning criterion, we introduce the concept of conditional linear weight, defined as follows.

**Definition 7 (Conditional Linear Weight).** *The conditional linear weight $w_{\mathcal{C}}$ is defined as the negative binary logarithm of the squared conditional linear correlation $\mathbb{C}_{\mathcal{C}}$ plus the number of conditions in the condition set $\mathcal{C}$, that is,*

$$w_{\mathcal{C}} := -\log_2 \mathbb{C}_{\mathcal{C}}^2 + |\mathcal{C}|.$$

*The optimal conditional linear weight of an S-box $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$ is defined as*

$$\mathcal{L}_{\mathcal{C}}(S) := \min_{\mathcal{C},(\lambda_i,\lambda_o)\in\mathbb{F}_2^{n+m}\setminus\{\mathbf{0}\}} w_{\mathcal{C}}(\lambda_i, \lambda_o),$$

*where the condition set $\mathcal{C}$ should go through all possible combinations of independent conditions.*

The conditional linear weight provides an evaluation of the resistance of a cipher against conditional linear cryptanalysis. Specifically, for a conditional linear trail with weight $w_{\mathcal{C}}$, approximately $2^{w_{\mathcal{C}}}$ samples are required to distinguish the cipher from a random permutation by exploiting the corresponding conditional linear correlation. Therefore, it offers a criterion for determining the optimal number of conditions and the corresponding optimal linear approximations.

*Complexity.* The conditional linear weight requires searching for the optimal correlations under all possible combinations of conditions. In theory, all possible CLATs need to be computed. However, as the number of conditions increases, the value $|\mathcal{C}|$ also increases, and the computation can be terminated once $|\mathcal{C}|$ exceeds the already derived optimal linear weight. For example, for the AES S-box, the optimal conditional linear weight is 3.83 when considering up to three conditions. When the number of conditions reaches four or more, the linear weight exceeds 3.83 according to Definition 7. Therefore, it is unnecessary to compute CLATs for cases with four or more conditions. The complexity of deriving the optimal conditional linear weight for the AES S-box is upper bounded by $\mathcal{O}(2^{5n+m}) = \mathcal{O}(2^{48})$, which can be completed on a PC within a few hours. Similarly, for the Midori S-box, when the number of conditions is one, the conditional linear weight is one, and we need not consider the other cases. Thus, the complexity of computing the optimal conditional linear weight for the Midori S-box is upper bounded by $\mathcal{O}(2^{3n+m}) = \mathcal{O}(2^{32})$.

In an SPN block cipher, the S-layer typically consists of parallel small S-boxes. Accordingly, the input masks, output masks, and condition masks can be partitioned into S-box-wise components.

**Proposition 4 (Independent S-boxes in Parallel).** *Given an $n$-bit S-layer composed of $t$ smaller S-boxes $S_0(x_0), S_1(x_1), \ldots, S_{t-1}(x_{t-1})$ with bit sizes $n_0$, $n_1$, …, $n_{t-1}$, respectively, i.e.,*

$$S(x) = (S_0(x_0)||S_1(x_1)||\ldots||S_{t-1}(x_{t-1})),$$

*where the $n$-bit input $x$ is split into $t$ $n_i$-bit ($n = \sum_i n_i$) disjoint parts as $x = (x_0||x_1||\ldots||x_{t-1})$. Similarly, the input and output linear masks $u$ and $v$ are split as $u = (u_0||u_1||\ldots||u_{t-1})$ and $v = (v_0||v_1||\ldots||v_{t-1})$, respectively. Then*

$$w_{\mathcal{C}}(u, v) = \sum_{i=0}^{t-1} w_{\mathcal{C}_i}(u_i, v_i),$$

*where $\mathcal{C} = \bigcup_{i=0}^{t-1} \mathcal{C}_i$, and $\mathcal{C}_i$ is the condition set for the S-box $S_i(x_i)$.*

*Proof.* Since the sub-S-boxes $S_0(x_0), S_1(x_1), \ldots, S_{t-1}(x_{t-1})$ are applied to disjoint subsets of input bits, the linear approximations of different $S_i$'s are mutually independent. In addition, the conditions in $\mathcal{C}_i$ only impose constraints on the input and/or the output of $S_i(x_i)$, and so the condition sets $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_{t-1}$ are mutually independent. Thus, we have

$$\begin{aligned} w_{\mathcal{C}}(u, v) &= -\log_2 \left(\mathbb{C}_{\mathcal{C}}(u, v)\right)^2 + |\mathcal{C}| = -\log_2 \left(\prod_{i=0}^{t-1} \mathbb{C}_{\mathcal{C}_i}(u_i, v_i)\right)^2 + \sum_{i=0}^{t-1} |\mathcal{C}_i| \\ &= -\log_2 \left(\prod_{i=0}^{t-1} \mathbb{C}^2_{\mathcal{C}_i}(u_i, v_i)\right) + \sum_{i=0}^{t-1} |\mathcal{C}_i| = -\sum_{i=0}^{t-1} \log_2 \mathbb{C}^2_{\mathcal{C}_i}(u_i, v_i) + \sum_{i=0}^{t-1} |\mathcal{C}_i| \\ &= \sum_{i=0}^{t-1} \left(-\log_2 \mathbb{C}^2_{\mathcal{C}_i}(u_i, v_i) + |\mathcal{C}_i|\right) = \sum_{i=0}^{t-1} w_{\mathcal{C}_i}(u_i, v_i). \end{aligned}$$

□

### 3.2 The Properties of the AES S-box

As the S-box of AES is widely used as the main nonlinear operation in many symmetric ciphers, we experimentally computed all its CLATs. The optimal conditional correlations and linear weights for different numbers of conditions are presented in Table 4. Furthermore, we propose the following experimental observation.

**Observation 1** *For the S-box of AES, given any non-zero output mask $\lambda_o \in \mathbb{F}_2^8$, there always exists a condition set $\mathcal{C}$ consisting of three independent conditions and an input mask $\lambda_i \in \mathbb{F}_2^8$ such that the conditional correlation $\mathbb{C}_{\mathcal{C}}(\lambda_i, \lambda_o) =$*

**Table 4.** The optimal conditional correlations and linear weights of the AES S-box under different numbers of conditions.

| $\#\mathcal{C}$ | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| $\lvert\mathbb{C}_{\mathcal{C}}\rvert$ | $2^{-3}$ | $2^{-2}$ | $2^{-1}$ | $2^{-0.415}$ | $2^{0}$ |
| $w_{\mathcal{C}}$ | 6 | 5 | 4 | **3.83** | 4 |

**Table 5.** Conditional correlations of the AES S-box under condition masks $0x0E$, $0x40$, and $0x87$ with linear mask pair $\lambda = (0x01, 0x01)$.

| Condition value | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| $\lambda$ holds | 16 | 16 | 14 | 28 | 17 | 17 | 16 | 16 |
| $\lambda$ fails | 16 | 16 | 18 | 4 | 15 | 15 | 16 | 16 |
| $\mathbb{C}_{\mathcal{C}}$ | 0 | 0 | $-2^{-3}$ | $2^{-0.415}$ | $2^{-4}$ | $2^{-4}$ | 0 | 0 |

$2^{-0.415}$ *and the conditional linear weight* $w_{\mathcal{C}} = 3.83$. *In other words, the optimal conditional linear weight of the AES S-box is achievable given any output mask. Moreover, for any output mask, the optimal linear weights in the five columns of Table 4 can always be satisfied under different numbers of conditions. These observations also hold for any input linear mask.*

For example, when $\lambda_i = 0x01$ and $\lambda_o = 0x01$, the correlation of the standard linear approximation of $\lambda_i x \oplus \lambda_o S(x)$ is $\mathbb{C} = 2^{-3.415}$. Now we apply three condition masks $0x0E$, $0x40$, and $0x87$ to the input $x$ and recompute the conditional correlations under different condition values. The results are shown in Table 5. It can be observed that when the condition values are 011, i.e., the condition set $\{(0x0E, 0), (0x40, 1), (0x87, 1)\}$ is satisfied, the conditional correlation reaches its maximum at $2^{-0.415}$.

Compared to the linear weight of 6, the conditional linear weight of 3.83 offers significant advantages by exploiting more properties of the AES S-box that are not revealed in standard linear cryptanalysis. Such properties motivate the necessity of reevaluating the security of SPN ciphers against linear cryptanalysis. Based on Observation 1, one can easily append one conditional round both before and after a standard linear distinguisher, and the conditional rounds can always achieve the optimal conditional correlations by filtering the plaintext and ciphertext. However, conditional linear trails derived from such an extension might not be optimal, and we propose a new automatic search model to find high-quality conditional linear trails in Section 4.2.

## 4 Generalized Framework for Conditional Linear Cryptanalysis

In this section, we present a new framework for conditional linear cryptanalysis based on the proposed CLAT. We begin by constructing conditional linear trails

and establishing an important theorem for conditional correlations. Based on this theoretical foundation, we introduce an MILP model for the automatic search of high-quality conditional linear trails. Finally, we describe how these trails can be used to mount distinguishing and key recovery attacks.

### 4.1 Construction of Conditional Linear Trails

We first present a theorem for conditional linear correlations over three parts. This theorem serves as the theoretical basis of our framework.

**Theorem 1.** *Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^p$, $G : \mathbb{F}_2^p \to \mathbb{F}_2^q$, and $H : \mathbb{F}_2^q \to \mathbb{F}_2^m$ be three vectorial Boolean functions. Let $X : \mathbb{F}_2^n \to \mathbb{F}_2^r$ and $Y : \mathbb{F}_2^m \to \mathbb{F}_2^s$ also be two vectorial Boolean functions. For $a \in \mathbb{F}_2^r$ and $b \in \mathbb{F}_2^s$, denote*

$$\mathcal{P}_a = \{x \in \mathbb{F}_2^n : X(x) = a\} \text{ and } \mathcal{Q}_b = \{z \in \mathbb{F}_2^q : Y(H(z)) = b\}.$$

*Moreover, denote the set of data satisfying the two condition sets by*

$$\mathcal{S}_{a,b} = \{x \in \mathcal{P}_a : G(F(x)) \in \mathcal{Q}_b\}.$$

*Assume that $\mathcal{P}_a$, $\mathcal{Q}_b$, and $\mathcal{S}_{a,b}$ are nonempty. Then, for any input mask $u \in \mathbb{F}_2^n$ and output mask $v \in \mathbb{F}_2^m$, we have*

$$\mathbb{C}^{H\circ G\circ F}\big|_{\mathcal{S}_{a,b}}(u,v) = \frac{|\mathcal{P}_a|\cdot|\mathcal{Q}_b|}{2^q\cdot|\mathcal{S}_{a,b}|}\sum_{\alpha\in\mathbb{F}_2^p,\beta\in\mathbb{F}_2^q}\mathbb{C}^H\big|_{\mathcal{Q}_b}(\beta,v)\cdot\mathbb{C}^G(\alpha,\beta)\cdot\mathbb{C}^F\big|_{\mathcal{P}_a}(u,\alpha). \tag{1}$$

*Proof.* The proof of this theorem refers to Appendix A. □

*Remark 2.* The coefficient factor in Formula (1),

$$\frac{|\mathcal{P}_a|\cdot|\mathcal{Q}_b|}{2^q\cdot|\mathcal{S}_{a,b}|} = \frac{|\mathcal{Q}_b|/2^q}{|\mathcal{S}_{a,b}|/|\mathcal{P}_a|},$$

characterizes the influence between the two imposed conditions which can be explained as follows.

- The numerator $|\mathcal{Q}_b|/2^q$ is the probability that a uniformly distributed element of $\mathbb{F}_2^q$ belongs to $\mathcal{Q}_b$. Equivalently, it is the probability that the output condition $Y(H(z)) = b$ is satisfied when the input $z$ of $H$ is uniformly distributed over $\mathbb{F}_2^q$.
- The denominator $|\mathcal{S}_{a,b}|/|\mathcal{P}_a|$ is the probability that $G(F(x)) \in \mathcal{Q}_b$ given $x \in \mathcal{P}_a$. In other words, it is the probability that the output condition holds when the input is restricted to $\mathcal{P}_a$.

Therefore, this coefficient factor measures how the input condition affects the probability of satisfying the output condition after propagation through $G \circ F$. When $\frac{|\mathcal{Q}_b|}{2^q} = \frac{|\mathcal{S}_{a,b}|}{|\mathcal{P}_a|}$, the output condition is independent of the input condition.

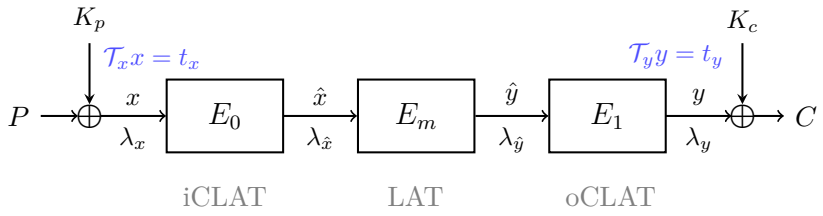



**Fig. 1.** The schematic of a conditional linear trail.

Theorem 1 can be used to build a three-part construction of conditional linear trails. Similar to the sandwich framework [12] in boomerang attacks and the DLCT framework [3] in differential-linear cryptanalysis, we decompose an iterative block cipher $E : \mathbb{F}_2^n \to \mathbb{F}_2^n$ into a cascade of three sub-ciphers, i.e.,

$$E = E_1 \circ E_m \circ E_0,$$

where the two boundary parts $E_0$ and $E_1$ are evaluated by conditional correlations, and the middle part $E_m$ is evaluated by standard linear correlations. The overall structure of a conditional linear trail is illustrated in Figure 1.

As shown in Figure 1, let $x$, $\hat{x}$, $\hat{y}$, and $y$ denote the input of $E_0$, the output of $E_0$, the input of $E_1$, and the output of $E_1$, respectively. Let

$$\mathcal{P} = \{x \in \mathbb{F}_2^n : \mathcal{T}_x \cdot x = t_x\} \text{ and } \mathcal{Q} = \{\hat{y} \in \mathbb{F}_2^n : \mathcal{T}_y \cdot E_1(\hat{y}) = t_y\}.$$

Furthermore, let

$$\mathcal{S} = \{x \in \mathcal{P} : E_m(E_0(x)) \in \mathcal{Q}\}.$$

By Theorem 1, given a linear approximation $(\lambda_x, \lambda_y)$, the conditional correlation over the data set $\mathcal{S}$ is

$$\mathbb{C}_{\mathcal{S}}(\lambda_x, \lambda_y) = \kappa \sum_{(\lambda_{\hat{x}}, \lambda_{\hat{y}}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n} \mathbb{C}_{\mathcal{P}}^{E_0}(\lambda_x, \lambda_{\hat{x}}) \cdot \mathbb{C}^{E_m}(\lambda_{\hat{x}}, \lambda_{\hat{y}}) \cdot \mathbb{C}_{\mathcal{Q}}^{E_1}(\lambda_{\hat{y}}, \lambda_y), \tag{2}$$

where $\kappa = \frac{|\mathcal{Q}|/2^n}{|\mathcal{S}|/|\mathcal{P}|}$. If the probability of data satisfying the output-side conditions is not significantly affected by the input-side conditions, i.e.,

$$\Pr_{x \in \mathcal{P}}(E_m(E_0(x)) \in \mathcal{Q}) \approx \frac{|\mathcal{Q}|}{2^n},$$

then we have $\kappa \approx 1$. Moreover, if the correlation of the conditional linear hull is dominated by a single linear trail with specific intermediate masks $(\lambda_{\hat{x}}, \lambda_{\hat{y}})$, the conditional correlation can be estimated as

$$\mathbb{C}_{\mathcal{S}}(\lambda_x, \lambda_y) \approx \mathbb{C}_{\mathcal{P}}^{E_0}(\lambda_x, \lambda_{\hat{x}}) \cdot \mathbb{C}^{E_m}(\lambda_{\hat{x}}, \lambda_{\hat{y}}) \cdot \mathbb{C}_{\mathcal{Q}}^{E_1}(\lambda_{\hat{y}}, \lambda_y). \tag{3}$$

Thus, for ciphers that satisfy the two assumptions mentioned above, Formula (3) can be used to efficiently estimate the conditional linear correlation. We experimentally validate the estimates given by Formula (3) for several AES-like ciphers in Section 5.2.

**Choice of the Boundary Rounds $E_0$ and $E_1$.** In our framework, we mainly consider the case where $E_0$ and $E_1$ each cover a single boundary round. In this setting, we can check if the plaintext/ciphertext satisfies the predefined conditions either directly or by guessing a small number of subkey bits. If $E_0$ or $E_1$ spans multiple rounds, one needs to guess many more subkey bits corresponding to those rounds, significantly increasing the filtering cost and overall attack complexity. We have also explored the use of multiple conditional rounds in our experiments, but the results have not been encouraging for the ciphers we examined. This is largely due to the strong diffusion properties of these ciphers, which, as noted, involve guessing a substantially larger number of subkey bits.

### 4.2 Automatic Search for Conditional Linear Trails

We now turn from the theoretical construction of conditional linear trails to their automatic search. Since the middle part $E_m$ is handled by standard linear propagation, the main new modeling ingredients lie in the two boundary components $E_0$ and $E_1$. We therefore first introduce two one-dimensional tables, namely the input-side CLAT and the output-side CLAT, which summarize the best conditional linear weights available at the two boundaries.

**Definition 8 (Input-side CLAT (iCLAT)).** *For a vectorial Boolean function $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$, the iCLAT of $S$ is a $2^m$-dimensional table, where each element stores the best conditional linear weight corresponding to each output mask $\lambda_o$, i.e., the entry at index $\lambda_o$ is*

$$\min_{\mathcal{C},\lambda_i} w_{\mathcal{C}}(\lambda_i, \lambda_o).$$

**Definition 9 (Output-side CLAT (oCLAT)).** *For a vectorial Boolean function $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$, the oCLAT of $S$ is a $2^n$-dimensional table, where each component stores the best conditional linear weight corresponding to each input mask $\lambda_i$, i.e., the entry at index $\lambda_i$ is*

$$\min_{\mathcal{C},\lambda_o} w_{\mathcal{C}}(\lambda_i, \lambda_o).$$

*Example 1.* Let $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$ be a bijective Boolean function as defined in Table 6. Its iCLAT and oCLAT tables are as below

$$\begin{aligned}
\text{iCLAT} = [&0, 1.385, 1.83, 1.83, 1.83, 1.83, 1.385, 1.83, \mathbf{1}, 1.83, 1.385, 1.385,\\
&1.385, 1.83, 1.83, 1.83, 1.83, 1.83, 1.83, 1.385, 1.83, 1.83, 1.83,\\
&1.385, 1.385, 1.385, 1.83, 1.385, 1.83, 1.385, 1.385, 1.385],\\
\text{oCLAT} = [&0, 1.385, 1.83, 1.83, 1.83, 1.83, 1.83, 1.385, \mathbf{1.385}, 1.83, 1.385,\\
&1.385, 1.83, 1.385, 1.385, 1.83, 1.83, 1.385, 1.83, 1.83, 1.83,\\
&1.385, 1.83, 1.83, 1.385, 2, 1.385, 1.385, 1.385, 1.83, 1.83, 1.83].
\end{aligned}$$

It can be observed that the minimum values, excluding the leading zero, in iCLAT and oCLAT are different. Therefore, when searching for active S-box

**Table 6.** S-box used in Example 1.

| $x$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $S(x)$ | E | 1D | 1 | 9 | 1B | 19 | 8 | 1C | 10 | 13 | 12 | 3 | B | 6 | 1F | 14 |
| $x$ | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
| $S(x)$ | 17 | 15 | 11 | 2 | 4 | 16 | 0 | 1A | F | C | 5 | 7 | 1E | D | A | 18 |

trails, it is necessary to consider the conditional linear weights on the input and output sides separately.

The search for conditional linear trails follows the three-part cascade that sequentially integrates iCLAT, LAT, and oCLAT to align with the cipher decomposition $E = E_1 \circ E_m \circ E_0$. Accordingly, the search integrates three different tables or propagation models:

- **iCLAT for $E_0$.** The iCLAT precomputes the optimal conditional linear weight for each output mask of $E_0$.
- **LAT for $E_m$.** The LAT is used to model standard linear propagation through the middle part $E_m$.
- **oCLAT for $E_1$.** The oCLAT precomputes the optimal conditional linear weight for each input mask of $E_1$.

In practice, we precompute the iCLAT and oCLAT for the nonlinear operations in each boundary round, and then use Proposition 4 to derive the optimal input or output masks and the corresponding linear weights for $E_0$ and $E_1$.

A simple way to construct a conditional linear trail is to extend a standard linear trail by one conditional round at each side. However, such a direct extension does not necessarily yield a high conditional correlation, since the route of an optimal conditional linear trail may differ substantially from that of an optimal classical linear trail. To address this issue, we propose a new MILP-based search strategy. We first search for promising byte-level trails that can benefit from conditional linear correlations, and then refine these byte-level templates into bitwise conditional linear trails.

**Searching for Byte Trails.** We mainly build upon the MILP modeling in [19] to search for byte trails. We follow the three-part framework that decomposes the target cipher $E$ into $E = E_1 \circ E_m \circ E_0$. Sub-ciphers $E_0$ and $E_1$ utilize conditional linear approximations and we use the optimal conditional linear weights $\mathcal{L}_{\mathcal{C}}(S)$ to indicate the contribution of one S-box to the overall correlation. The middle sub-cipher $E_m$ employs standard linear approximations and uses the optimal linear weights $\mathcal{L}(S)$. The objective of the search is to minimize the total linear weight of the whole conditional linear trail, which is formulated as

$$\min \quad \sum_{E_0} A_t \cdot \mathcal{L}_{\mathcal{C}}(S) + \sum_{E_m} A_t \cdot \mathcal{L}(S) + \sum_{E_1} A_t \cdot \mathcal{L}_{\mathcal{C}}(S). \tag{4}$$

It can be seen that such an objective function can provide a bound on the correlation of a conditional linear trail.

**Searching for Bitwise Trails.** Based on the derived byte templates above, we further utilize the method in [33] to model the nonlinear and linear operations at the bit level and search for bitwise trails. The critical difference of the modeling from [33] is the objective function, where the linear weights of S-boxes in $E_0$ and $E_1$ are replaced by the optimal conditional linear weights from iCLAT and oCLAT, respectively.

The objective function is formulated as

$$\min \sum_{E_0, A_t=1} w_{\text{iCLAT}}[t] + \sum_{E_m, A_t=1} w_{\text{LAT}}[t] + \sum_{E_1, A_t=1} w_{\text{oCLAT}}[t], \tag{5}$$

where $w_{\text{iCLAT}}, w_{\text{LAT}}$, and $w_{\text{oCLAT}}$ are the actual linear weights of S-boxes in $E_0, E_m$, and $E_1$, respectively. We note that for AES, when $E_0$ and $E_1$ only involve one round each, $w_{\text{iCLAT}}$ and $w_{\text{oCLAT}}$ can always achieve the optimal values according to Observation 1.

### 4.3 Distinguishing Attacks Based on Conditional Linear Trails

We next describe how to use a conditional linear trail to distinguish a target cipher from a random permutation.

**The Basis for Distinguishing Attacks.** For a block cipher $E$ with a block length of $n$, suppose there exists a conditional linear approximation $(\lambda_i, \lambda_o)$ under the condition set $\mathcal{C} = \{(\mathcal{T}_0, t_0), \ldots, (\mathcal{T}_{c-1}, t_{c-1})\}$, where each $\mathcal{T}_i \in \{0,1\}^{2n}$, that exhibits a conditional correlation $\mathbb{C}$.

If no whitening keys are applied before the first round or after the last round, we can directly discard plaintexts and ciphertexts based on these conditions and use the remaining data to launch a distinguishing attack. However, when whitening keys are present, these keys transform one data subspace into another, and the resulting data subspace may not correspond to the one required for the nonlinear operations.

Therefore, instead of discarding data, we partition it into $2^c$ subspaces according to the condition values. Data in one of these subspaces will satisfy the conditions for the nonlinear components, introducing a notable correlation $\mathbb{C}$ that can be detected. In contrast, for a random permutation $R$, the correlation of any linear approximation within any subspace is expected to be zero. Thus, verifying the existence of a data subspace that yields a significantly higher correlation for the linear approximation enables us to distinguish the target cipher from a random permutation.

**The Primary Steps for Distinguishing Attacks.** The distinguishing attack based on conditional linear trails can be divided into the following three steps.

1. **Data Partition**. Classify the plaintext-ciphertext pairs according to the condition set $\mathcal{C}$, ensuring that each data subspace $D_i$ contains a sufficiently large number of plaintext-ciphertext pairs $\{(P, C)\}$, where $C = E(P)$.

**Algorithm 1** Distinguishing attack based on the conditional linear trail

**Input:** Linear mask pair $(\lambda_i, \lambda_o)$, a condition set $\mathcal{C} = \{(\mathcal{T}_j, t_j)\}$, a threshold $\tau$, and $N$ plaintext-ciphertext pairs $(P, C)$.
**Output:** The target cipher $E$ or a random permutation $R$.

$c \leftarrow |\mathcal{C}|$
Initialize two types of $2^c$ counters, $T_1[x_0, x_1, \ldots, x_{c-1}]$ and $T_2[x_0, x_1, \ldots, x_{c-1}]$, where $x_i \in \mathbb{F}_2$.
**for** $i$ **from** 0 **to** $N-1$ **do**
    **for** $j$ **from** 0 **to** $c-1$ **do**
        $x_j \leftarrow \mathcal{T}_j \cdot (P_i, C_i)$
    $T_1[x_0, x_1, \ldots, x_{c-1}] \leftarrow T_1[x_0, x_1, \ldots, x_{c-1}] + (\lambda_i \cdot P_i \oplus \lambda_o \cdot C_i)$
    $T_2[x_0, x_1, \ldots, x_{c-1}] \leftarrow T_2[x_0, x_1, \ldots, x_{c-1}] + 1$
**for** $(x_0, x_1, \ldots, x_{c-1})$ **from** $(0, 0, \ldots, 0)$ **to** $(1, 1, \ldots, 1)$ **do**
    $\hat{\mathbb{C}}_{(x_0, x_1, \ldots, x_{c-1})} \leftarrow |1 - 2 \times T_1[x_0, x_1, \ldots, x_{c-1}]/T_2[x_0, x_1, \ldots, x_{c-1}]|$
    **if** $\hat{\mathbb{C}}_{(x_0, x_1, \ldots, x_{c-1})} > \tau$ **do**
        **Return** $E$.
**Return** $R$.

2. **Correlation Computation**. For each data subspace $D_i$, compute the empirical conditional correlation $\hat{\mathbb{C}}_i$ of the linear approximation using the formula below

$$\hat{\mathbb{C}}_i = \frac{|\{(P, C) \in D_i \mid \lambda_i P \oplus \lambda_o C = 0\}| - |\{(P, C) \in D_i \mid \lambda_i P \oplus \lambda_o C = 1\}|}{N_i},$$

   where $N_i$ represents the number of data in $D_i$.
3. **Hypothesis Testing**. Set the significance threshold $\tau$ based on the statistical significance level (e.g., 95% confidence). If there exists a data subset $D_i$ satisfying $|\hat{\mathbb{C}}_i| > \tau$, the permutation is classified as the target cipher $E$; otherwise, the permutation is considered as a random permutation $R$.

In Algorithm 1, we present the pseudocode for the distinguishing attack, which utilizes two types of counters of size $2^c$ to compute the conditional correlation in each data subspace. Consequently, the memory complexity of the distinguishing attack is $\mathcal{O}(2^{c+1})$, while both the time and data complexities are $N$. Next, we provide a detailed analysis of the data complexity $N$.

**Data Complexity.** Since empirical correlations are computed in all $2^c$ data subspaces, and the maximum observed correlation is expected to occur precisely under the correct key hypothesis, we require the attack to achieve a $c$-bit advantage. Following the statistical model in [32], if the correlation of a linear approximation is $\mathbb{C}_{\mathcal{C}}$, the number of plaintext-ciphertext pairs required to gain a $c$-bit advantage with success probability $P_s$ is

$$N_{\mathcal{C}} = \frac{\left(\Phi^{-1}(P_s) + \Phi^{-1}(1 - 2^{-c-1})\right)^2}{\mathbb{C}_{\mathcal{C}}^2}, \tag{6}$$

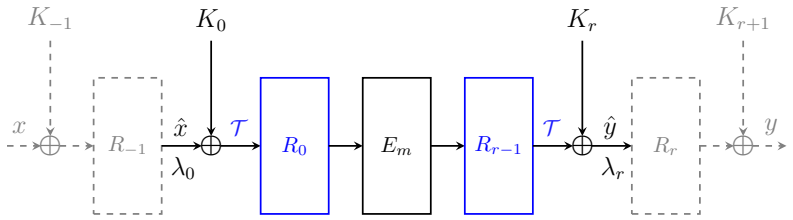



**Fig. 2.** Two settings of key recovery attacks based on conditional linear trails.

where $\Phi$ is the cumulative distribution function of the standard normal distribution. The data volume $N_{\mathcal{C}}$ corresponds to the required amount of data that satisfies the specific condition set $\mathcal{C}$. In the context of a known-plaintext attack, it is necessary to obtain at least $N_{\mathcal{C}}$ samples for each subspace. As stated in [1], this scenario can be modeled as the dixie-cup problem that requires obtaining at least $N_{\mathcal{C}}$ data samples for each of the $2^c$ types. Newman provided a formula for deriving the total required number $N$ of samples to achieve that in [30], that is,

$$N = 2^c \cdot \ln(2^c) + (N_{\mathcal{C}} - 1) \cdot 2^c \ln(\ln(2^c)) + \mathcal{O}(1). \tag{7}$$

Therefore, a successful distinguishing attack can be achieved with probability $P_s$ when the data complexity is $N$ using the conditional linear trail.

### 4.4 Key Recovery Attacks Based on Conditional Linear Trails

We now describe how conditional linear trails can be used to launch key recovery attacks. Based on the number of active S-boxes in the conditional rounds, we propose two scenarios for key recovery. For clarity, active S-boxes in the conditional rounds are referred to as *conditional active S-boxes*. In the following, we assume that the condition set $\mathcal{C} = \{(\mathcal{T}_0, t_0), \ldots, (\mathcal{T}_{c-1}, t_{c-1})\}$ is given.

**Key Recovery Attacks Derived from Distinguishing Attacks.** The first scenario is a straightforward application of the distinguishing attack described in Section 4.3. The key recovery model is depicted by the solid lines in Figure 2. Note that the condition masks $\mathcal{T}$ are applied to $(P \oplus K_0)||(C \oplus K_r)$, where $P$ and $C$ are $\hat{x}$ and $\hat{y}$ in Figure 2, respectively. In the distinguishing attack, suppose that the condition values of the data subspace exhibiting the optimal conditional correlation are $(v_0, v_1, \ldots, v_{c-1})$. Then, we get

$$(\mathcal{T}_0 \cdot (K_0||K_r), \ldots, \mathcal{T}_{c-1} \cdot (K_0||K_r)) = (v_0 \oplus t_0, \ldots, v_{c-1} \oplus t_{c-1}).$$

Only those values of $K_0||K_r$ satisfying the above constraint are preserved and the exact value can be further recovered through auxiliary methods. Therefore, the complexity is consistent with that of the distinguishing attack.

**Key Recovery Attacks via Guessing Additional Keys.** When the number of conditional active S-boxes is small, we can extend the attack to $r+1$ or $r+2$ rounds by guessing additional subkeys. Take the case shown in Figure 2 as an example, where the dashed lines indicate extending the attack by one round forward and one round backward, and we denote these rounds as $R_{-1}$ and $R_r$, respectively. Besides, the middle $r$ rounds, shown with solid lines in Figure 2, are characterized by conditional linear trails.

As shown in Figure 2, let $\hat{x}$ and $\hat{y}$ denote the internal states at the input and output of the middle conditional linear trail, respectively. Here, $\hat{x}$ is obtained by partially encrypting the plaintext through $R_{-1}$, and $\hat{y}$ is obtained by partially decrypting the ciphertext through $R_r$. We do not need full knowledge of $\hat{x}$ and $\hat{y}$, and it suffices to extract their relevant parts, namely, the masked values $\mathcal{T}\cdot(\hat{x}\|\hat{y})$, $\lambda_0\cdot\hat{x}$, and $\lambda_r\cdot\hat{y}$, where $\lambda_0$ and $\lambda_r$ are the input and output masks of the middle conditional linear trail, respectively. The subkey bits of $K_{-1}$ and $K_{r+1}$ that affect these masked values are denoted by $k_{-1}$ and $k_{r+1}$.

For each guess of $k_{-1}$, $k_{r+1}$, and $\mathcal{T}\cdot(k_0\|k_r)$, we partially encrypt and decrypt each plaintext-ciphertext pair to obtain the relevant parts of $\hat{x}$ and $\hat{y}$. We then compute the conditional value vector $\mathcal{T}\cdot(\hat{x}\|\hat{y})\oplus\mathcal{T}\cdot(k_0\|k_r)$. Only data satisfying the predefined conditions are retained, and the empirical conditional correlation is computed over this filtered subset. When all three guesses are correct, this conditional correlation is expected to be non-negligible, which allows us to identify the correct key bits. A direct implementation of the attack requires time complexity $N\cdot 2^{|k_{-1}|+|k_{r+1}|+c}$, but it can be accelerated using the fast Walsh-Hadamard transform (FWHT), as described below.

For each value of $\mathcal{T}\cdot(k_0||k_r)$, we define a matrix $\boldsymbol{B}$ of size $2^{|k_{-1}|+|k_{r+1}|}\times 2^{|k_{-1}|+|k_{r+1}|}$, with rows and columns indexed by $i$ and $j=k_{-1}||k_{r+1}$, respectively, where $i$ denotes the $(|k_{-1}|+|k_{r+1}|)$-bit value obtained by restricting $x||y$ to the active bit positions in $R_{-1}$ and $R_r$. Specifically, $B_{ij}$ is defined as follows:

$$B_{ij}=\begin{cases}1, & \text{if } \mathcal{T}\cdot(\hat{x}||\hat{y})\oplus\mathcal{T}\cdot(k_0||k_r)=t \text{ and } \lambda_0\cdot\hat{x}\oplus\lambda_r\cdot\hat{y}=0;\\ -1, & \text{if } \mathcal{T}\cdot(\hat{x}||\hat{y})\oplus\mathcal{T}\cdot(k_0||k_r)=t \text{ and } \lambda_0\cdot\hat{x}\oplus\lambda_r\cdot\hat{y}=1;\\ 0, & \text{if } \mathcal{T}\cdot(\hat{x}||\hat{y})\oplus\mathcal{T}\cdot(k_0||k_r)\neq t.\end{cases} \tag{8}$$

Let $L$ denote the column vector of size $2^{|k_{-1}|+|k_{r+1}|}$, where $L[i]$ counts the collected plaintext-ciphertext pairs whose bits at the active positions of $R_{-1}$ and $R_r$ are equal to $i$. Then given a fixed guess of $\mathcal{T}\cdot(k_0||k_r)$, the vector of empirical conditional correlations over all guesses of $k_{-1}||k_{r+1}$ is given by $\boldsymbol{B}\cdot L$. We note that the matrix $\boldsymbol{B}$ possesses the circulant structure proposed in [7], as the values of $\hat{x}$ and $\hat{y}$ are uniquely determined by $R_{-1}(x\oplus k_{-1})||R_r^{-1}(y\oplus k_{r+1})$. Therefore, we can use the fast Walsh-Hadamard transform to accelerate the key recovery process as described below.

1. **Data Distillation**. Initialize a column vector $L$ of size $2^{|k_{-1}|+|k_{r+1}|}$ by zeros. For each plaintext-ciphertext pair $(P,C)$, compute the $(|k_{-1}|+|k_{r+1}|)$-bit value $i$ corresponding to the active bit positions in $R_{-1}$ and $R_r$. Increase $L[i]$ by one.

2. **Derive the First Column of the Circulant Matrix**. As $\boldsymbol{B}$ is a circulant matrix, we only need to derive the first column (or row) to obtain the full matrix. For each condition value $t$ of the inner key bits $k_0||k_r$, i.e., $t = \mathcal{T} \cdot (k_0||k_r)$, where $0 \leq t \leq 2^c - 1$ and $c$ is the number of conditions in the condition set:
   (a) We construct a vector $E_t$ of size $2^{|k_{-1}|+|k_{r+1}|}$ and initialize it by zeros.
   (b) For each value $v$ of the combination of $k_{-1}$ and $k_{r+1}$:
      i. We use it to partially encrypt $x$ and decrypt $y$, deriving parts of $\hat{x}$ and $\hat{y}$, respectively, where $\hat{x}||\hat{y} = R_{-1}(x \oplus k_{-1})||R_r^{-1}(y \oplus k_{r+1})$.
      ii. Compute the condition value $l = \mathcal{T} \cdot (\hat{x}||\hat{y})$.
      iii. Compute the value of $t \oplus l$ and decide the value of $E_t[v]$ according to Formula (8).
3. **FWHT**. Compute the Fourier transforms of $L$ and $E_t$ using FWHT and denote them as $\mathcal{F}_L$ and $\mathcal{F}_{E_t}$, respectively, for $0 \leq t \leq 2^c - 1$. Compute $\mathcal{G}_t = \mathcal{F}_{E_t} \circ \mathcal{F}_L$, where $\circ$ denotes the point-wise multiplication, and perform the inverse FWHT of $\mathcal{G}_t$, which is denoted as $\mathcal{G}'_t$, for $0 \leq t \leq 2^c - 1$.
4. **Analysis.** Identify the largest entry among all vectors $\mathcal{G}'_0, \mathcal{G}'_1, \ldots, \mathcal{G}'_{2^c-1}$, say the entry at index $v$ of $\mathcal{G}'_t$. Then the values of the outer key bits $k_{-1}||k_{r+1}$ and $\mathcal{T} \cdot (k_0||k_r)$ are $v$ and $t$, respectively.

*Complexity.* Step 1 involves processing plaintext-ciphertext pairs, with the data complexity $N$ determined by Formulas (6) and (7). In Step 2, we need to construct $2^c$ vectors of size $2^{|k_{-1}|+|k_{r+1}|}$, yielding a complexity of $O(2^c \times 2^{|k_{-1}|+|k_{r+1}|})$. In Step 3, we perform $2^c + 1$ FWHT and $2^c$ inverse FWHT. The complexity of this step is $\mathcal{O}((2^{c+1} + 1) \times (|k_{-1}| + |k_{r+1}|) \times 2^{|k_{-1}|+|k_{r+1}|})$. The cost of Step 4 is negligible compared to other steps. Therefore, the total time complexity is dominated by $\mathcal{O}((2^{c+1} + 1) \times (|k_{-1}| + |k_{r+1}|) \times 2^{|k_{-1}|+|k_{r+1}|}) + \mathcal{O}(N)$. Since we need to construct $2^c + 1$ tables, each of size $2^{|k_{-1}|+|k_{r+1}|}$, the memory complexity is $\mathcal{O}((2^c + 1) \times 2^{|k_{-1}|+|k_{r+1}|})$.

*Experimental Verification.* To verify the correctness of the FWHT-based key recovery process, we perform experiments on a simple conditional linear approximation of Midori-128. Specifically, we use a two-round conditional linear trail with conditional correlation $2^{-3}$ and add one round at the front to recover the secret key bits. Using the FWHT, the correlation for the correct key guess attains the maximum. We present more details of the experiment in Appendix B, and the source code is available on GitHub.

## 5 Applications

In this section, we apply our generalized framework for conditional linear cryptanalysis to several AES variants and AES-like ciphers, including AES [10], Rijndael-256 [9], ARIA [23], LED [18], Midori-128 [2], and SKINNY-128 [4]. We first present new security bounds against linear cryptanalysis and improved linear distinguishers for these cryptographic primitives, and then propose key recovery attacks against AES, Rijndael-256, and Midori-128.

**Table 7.** The optimal (conditional) linear weights of some S-boxes.

| Cipher | AES | ARIA | LED | Midori | SKINNY |
|---|---|---|---|---|---|
| $\mathcal{L}(S)$ | 6 | 6 | 2 | 2 | 2 |
| $\mathcal{L}_{\mathcal{C}}(S)$ | 3.83 | 3.83 | 1 | 1 | 1 |

### 5.1 New Security Bounds Against Linear Cryptanalysis

Security bounds against linear cryptanalysis can be used to evaluate the resistance of a cipher to linear cryptanalysis. A typical way to derive these bounds when designing a new cipher involves multiplying the number of active S-boxes by the optimal linear weight of one S-box. However, these bounds are violated within our generalized framework for conditional linear cryptanalysis.

In Table 7, we present the optimal linear and conditional linear weights for the S-boxes employed in the target cryptographic algorithms. The results demonstrate significant improvements in optimal conditional linear weights over standard linear weights. This indicates that CLATs can reveal subspace statistical properties hidden within the full-space LATs, offering a new tool for evaluating the resistance of a cipher to linear cryptanalysis.

Next, we present several identified conditional linear trails whose weights are below the bounds obtained in the standard full-space linear-trail model for the considered ciphers. These results indicate that conventional bounds for standard linear cryptanalysis do not directly capture the behavior of conditional linear trails. The results are summarized in Table 1 in Section 1.1. In the table, "B" denotes a lower bound on the weight of a standard or conditional linear trail, obtained by multiplying the number of active S-boxes by the optimal weight of the underlying S-box. For conditional linear trails, the corresponding bound is computed according to Formula (4). The symbol "E" denotes the exact weight of a concrete trail identified by our search. Due to computational limitations, these identified trails may not be globally optimal. Nevertheless, they improve the currently known upper bounds on the minimum weights of conditional linear trails, thereby leading to refined security evaluations against conditional linear cryptanalysis. More details, including the search results and trail specifications, are available in our GitHub repository.

It should be emphasized that all trails reported in Table 1 are based on single standard or conditional linear trails. For ciphers where the correlation is dominated by a single trail, such as AES, LED, and Midori, as experimentally observed in Section 5.2, these weights provide a reliable estimate for evaluating resistance against standard or conditional linear distinguishers. However, for ciphers exhibiting a significant linear-hull effect, such as SKINNY-128, the signed sum of multiple trails may lead to an actual correlation that deviates substantially from the single-trail prediction. In such cases, the reported single-trail weights should not be interpreted as strict lower bounds on resistance against

linear or conditional linear cryptanalysis. Nevertheless, according to the results of [24], a single trail still characterizes the security for at least half of the keys.

Next, we present some specific results.

*AES.* For 4-round AES, Ashur and Takke reported a conditional linear trail with a conditional linear weight of 118 [1]. Using the CLAT, we derive the optimal 4-round conditional linear trail with a conditional linear weight of 106.6 and an associated conditional correlation of $2^{-23.3}$ under 60 conditions. The optimal conditional linear trail is illustrated in Figure 4.

*Rijndael-256.* For Rijndael-256, the linear weights of standard linear trails over four and five rounds are 150 and 246, respectively. By utilizing CLATs, we derive that the conditional linear weight for four rounds is 106.6, with a conditional correlation of $2^{-23.3}$ under 60 conditions. For five rounds, the conditional linear weight is 176.56 under 96 conditions, with a conditional correlation of $2^{-40.28}$.

*ARIA.* For 6-round ARIA, the lower bound on the linear weights of standard linear trails is 144, suggesting that no valid linear trails exist for six rounds. However, we present a conditional linear trail for 6-round ARIA with a conditional correlation of $2^{-51.32}$ under 24 conditions, which corresponds to a conditional linear weight of 126.64. Therefore, we increase the previous security bounds of ARIA against linear cryptanalysis from six to seven rounds in terms of a single linear trail.

*LED.* For LED, it was reported in [34] that no valid linear trails exist for six rounds. However, we identify a 6-round conditional linear trail with a conditional linear weight of 61, involving five conditions and exhibiting a conditional correlation of $2^{-28}$. Additionally, we increase the previous security bounds of LED against linear cryptanalysis from six to seven rounds based on a single linear trail.

*Midori-128.* For Midori-128, it was reported in [34] that no valid linear trails exist for 12 rounds. However, we identify a conditional linear trail with a conditional linear weight of 124, which involves 12 conditions and exhibits a conditional correlation of $2^{-56}$. Additionally, we increase the previous security bounds of Midori-128 against linear cryptanalysis from 12 to 13 rounds in terms of a single linear trail.

*SKINNY-128.* For SKINNY-128, we present a conditional linear trail for 14-round SKINNY-128 with a conditional correlation of $2^{-59.66}$ under six conditions, which corresponds to a conditional linear weight of 125.32. This is the first valid linear trail for 14-round SKINNY-128.

### 5.2 Experimental Verification of Conditional Linear Correlations

In this subsection, we experimentally verify the correlations of conditional linear approximations for several round-reduced AES-like ciphers. For each cipher, we

**Table 8.** Experimental verification of conditional linear correlations.

| Cipher | Round | Theoretical | | Experimental | | | | |
|---|---|---|---|---|---|---|---|---|
| | | #Cond. | Corr.* | Data | Filtered Data | Corr. | Mean | S.D. |
| AES | 3 | 24 | $2^{-6.32}$ | $2^{40}$ | $[2^{15.990}, 2^{16.016}]$ | $[2^{-7.257}, 2^{-5.579}]$ | $2^{-6.323}$ | $2^{-8.102}$ |
| LED | 4 | 20 | $2^{-5}$ | $2^{34}$ | $[2^{13.985}, 2^{14.026}]$ | $[2^{-6.154}, 2^{-4.287}]$ | $2^{-4.982}$ | $2^{-7.053}$ |
| Midori | 5 | 18 | $2^{-7}$ | $2^{37}$ | $[2^{18.995}, 2^{19.005}]$ | $[2^{-7.657}, 2^{-6.460}]$ | $2^{-7.030}$ | $2^{-9.572}$ |
| SKINNY | 6 | 11 | $2^{-8}$ | $2^{31}$ | $[2^{19.996}, 2^{20.003}]$ | $[2^{-14.48}, 2^{-3.05}]^{\dagger}$ | - | - |

S.D.: Standard deviation of empirical correlations over independent experiments
$*$: Correlation refers to the conditional correlation of a single trail.
†: Among 128 independent experiments, 76 produced correlations greater than $2^{-8.5}$, which is consistent with Theorem 2 in [24].

select a representative conditional linear trail found by automatic search tools and perform multiple independent experiments, each with a fresh key. In each experiment, we generate an independent set of plaintext-ciphertext pairs, filter them based on the predefined condition set, and compute the empirical conditional correlation for the associated linear approximation using the remaining data. The experiments are performed on 3-round AES, 4-round LED, 5-round Midori, and 6-round SKINNY. We omit experimental verification of the correlations of conditional linear approximations for ARIA due to its high complexity. Nevertheless, ARIA has a similar structure to AES and employs the same type of S-boxes, and thus its conditional correlation behavior is expected to be similar to that of AES.

The main results are summarized in Table 8 and more details of the experimental verification are provided in Appendix C. As shown in Table 8, the number of filtered plaintext-ciphertext pairs matches the expected filtering probability determined by the condition set. This supports the assumption that, for well-designed ciphers with strong diffusion and sufficient middle rounds, the factor $\kappa$ is close to one. Notably, this assumption is expected to become more plausible as the number of middle rounds increases.

For AES, LED, and Midori, the experimental results indicate that the corresponding approximations are dominated by a single trail. In these cases, Formula (3) provides an accurate prediction of the conditional correlation. In contrast, SKINNY exhibits a significant linear hull effect. For this cipher, the single-trail correlation does not fully determine the actual correlation; nevertheless, as suggested by Theorem 2 in [24], the single-trail estimate can still be used to evaluate security for at least half of the keys.

### 5.3 Distinguishing Attacks

Based on the identified conditional linear trails, we propose distinguishing attacks on these ciphers. Following the distinguishing attack procedure in Section 4.3 and using Formulas (6) and (7), we compute the data complexities of the attacks based on the numbers of conditions and conditional correlations. Table 9

**Table 9.** Comparison of conditional linear distinguishers with previous optimal (conditional) linear distinguishers. The success probabilities of these distinguishers are 99.8%.

| Cipher | Round | Type | $\mathbb{C}_{\mathcal{C}}$ | $\lvert\mathcal{C}\rvert$ | Weight | $N$ | Ref. |
|---|---|---|---|---|---|---|---|
| AES | 4 | Cond. Linear | $2^{-43}$ | 32 | 118 | $2^{126.04\dagger}$ | [1] |
| | | Cond. Linear | $2^{-23.3}$ | 60 | 106.6 | $\mathbf{2^{115.60}}$ | Our |
| Rijndael-256 | 5 | Linear | - | - | 246 | $2^{249}$ | [34] |
| | | Cond. Linear | $2^{-40.28}$ | 96 | 176.56 | $\mathbf{2^{186.28}}$ | Our |
| ARIA | 5 | Linear | - | - | 105.2 | $2^{108.2}$ | [34] |
| | | Cond. Linear | $2^{-47.43}$ | 6 | 100.86 | $\mathbf{2^{106.18}}$ | Our |
| LED | 5 | Linear | - | - | 58 | $2^{61}$ | [34] |
| | | Cond. Linear | $2^{-19}$ | 12 | 50 | $\mathbf{2^{56.50}}$ | Our |
| Midori-128 | 11 | Linear | - | - | 124 | $2^{127}$ | [34] |
| | | Cond. Linear | $2^{-54}$ | 8 | 116 | $\mathbf{2^{121.83}}$ | Our |
| SKINNY-128‡ | 13 | Linear | - | - | 122 | $2^{125}$ | [34] |
| | | Cond. Linear | $2^{-56.66}$ | 5 | 118.32 | $\mathbf{2^{123.30}}$ | Our |

† The complexity has been recomputed to achieve a 99.8% success probability.
‡ SKINNY exhibits a significant linear-hull effect; when only a single trail is considered, the attack is effective for at least half of the keys.

provides a comparison of our conditional linear distinguishers and existing optimal (conditional) linear distinguishers, under the attack success probability of 99.8%. The complexities of standard linear distinguishers are computed as $8\mathbb{C}^{-2}$ to achieve this targeted success probability, based on the results reported in [27].

We note that the standard linear distinguishers in Table 9 are derived from optimal linear trails, while the conditional linear trails for our conditional linear distinguishers may not be optimal, as we did not explore the entire search space. This suggests that conditional linear distinguishers could be further improved by expanding the search space and identifying better conditional linear trails.

### 5.4 Key Recovery Attacks on AES, Rijndael-256 and Midori-128

In this subsection, we present key recovery attacks on AES, Rijndael-256, and Midori-128 based on the conditional linear trails identified by our framework. All the attacks described below are conducted in the known-plaintext model.

*Key Recovery Attack on 4-Round AES.* In [1], Ashur and Takke presented a key recovery attack on 4-round AES using a conditional linear trail with time and data complexities of $2^{125.72}$. Based on the optimal conditional linear trail illustrated in Figure 4, we propose an improved linear key recovery attack on 4-round AES. According to Formulas (6) and (7), when the amount of data per subspace is $2^{53.38}$, the attack success probability is 95%, with a total data complexity of $2^{115.28}$. By applying the key recovery approach described in Section 4.4, the data

and time complexities for the 4-round key recovery attack are $2^{115.28}$, and the memory complexity is $2^{61}$. This represents an improvement by a factor of $2^{10.44}$ in complexity over the attack presented in [1].

*Key Recovery Attack on 7-Round Rijndael-256.* We present the first linear key recovery attack on 7-round Rijndael-256, based on a 5-round conditional linear trail. The conditional linear trail exhibits a conditional linear correlation of $2^{-97.42}$ under 36 conditions. The detailed attack diagram is provided in Appendix E. In total, there are 24 active S-boxes across these two rounds, corresponding to 192 key bits that need to be guessed. According to Formulas (6) and (7), when the data complexity reaches $2^{241.06}$, the success probability of recovering the correct key bits exceeds 95%. We note that the advantage in Formula (6) is set as $192 + 36 = 228$, as we need to additionally guess the 36-bit condition value. We then apply the key recovery procedure outlined in Section 4.4 to recover these key bits and the time complexity is computed as $(2^{37}+1) \times 192 \times 2^{192} + 2^{241.06} = 2^{241.12}$. The memory complexity is $2^{228}$.

*Key Recovery Attack on 12-Round Midori-128.* We present the first linear key recovery attack on 12-round Midori-128, based on a 10-round conditional linear trail. The detailed attack diagram is provided in Appendix F. The 10-round conditional linear trail involves four conditions and exhibits a conditional linear correlation of $2^{-52}$. As illustrated in Figure 6, we need to guess 96 key bits in the first and last rounds, along with the 4-bit condition value. According to Formulas (6) and (7), the success probability of recovering these 100 key bits is about 95% when the amount of data is $2^{115.47}$. We then apply the key recovery procedure in Section 4.4 to recover these key bits and the time complexity is $(2^5 + 1) \times 96 \times 2^{96} + 2^{115.47} = 2^{115.48}$. The memory complexity is $2^{100}$.

## 6 Discussions and Open Problems

### 6.1 Discussions

*New Valid Trails in Linear Cryptanalysis.* We note that our conditional linear trails are not derived by simply appending conditional rounds to standard linear trails; rather, they are obtained through new search processes from a global perspective. The new framework changes the search strategy for linear trails: traditional searches tend to look for routes with fewer active S-boxes, whereas the search for conditional linear trails allows for a greater number of active S-boxes, with more S-boxes located within $E_0$ and $E_1$. This expansion of the search space enables us to identify high-quality linear trails that are undetectable by standard linear cryptanalysis. This is one of the reasons why conditional linear cryptanalysis can surpass the security bounds of standard linear cryptanalysis.

*The Gap between (Conditional) Linear Weights and Complexities of (Conditional) Linear Distinguishers.* The (conditional) linear weight $w$ describes the strength of the statistical correlation of a linear approximation and directly

determines the data complexity of a distinguishing attack. However, given a required success probability, the actual data complexity is typically $2^w$ multiplied by a small constant. As shown in Table 9, when the success probability is set as 99.8%, the data complexities of standard linear distinguishing attacks are typically eight times $2^w$. However, for conditional linear attacks, the constants vary between $2^{5.0} \sim 2^{9.7}$ in this context. Generally, as the size of the condition set grows, the constant also increases. This is because a larger number of conditions requires identifying the correct subspace from more random cases, which increases the probability of error. On the other hand, when the size of the condition set grows, the improvement of the conditional linear distinguisher over the standard linear distinguisher becomes more distinct. This is because such a conditional linear distinguisher involves more conditional S-boxes, thereby introducing greater gain. For instance, the complexity of the conditional distinguishing attack against 5-round Rijndael-256 is $2^{186.28}$ under 96 conditions, improving upon the optimal standard linear distinguishing attack by a factor of $2^{62.72}$; while for 5-round ARIA under 6 conditions, the factor between the two complexities is only $2^{2.02}$.

*Linear vs. Nonlinear Condition Selection.* The general framework proposed in this paper primarily focuses on linear conditions, which means the filtered data are in specific subspaces. Theoretically, using nonlinear conditions to filter data within a general subset (not necessarily a subspace) is also applicable, which may introduce stronger conditional linear correlations and weights. The main reason for restricting our analysis to linear conditions is that round keys in block ciphers prevent direct data filtering, and one has to guess a number of key bits to enable data filtering. Under linear conditions, this guessing can be confined to linearly transformed key bits within the target linear subspace. In contrast, nonlinear conditions typically require explicit guessing of additional key bits to determine whether a sample belongs to the target data subset, which increases the overall complexity. Using general conditions, both linear and nonlinear, may provide advantages in some specialized settings when round keys are known, such as in [22], and we leave this for future work.

### 6.2 Open Problems

*More Efficient Data Usage.* Given a set of condition masks, our distinguisher employs only a single data subspace under a fixed set of condition values. However, in fact, we can always find good linear trails for different data subspaces that exhibit strong conditional correlations. This indicates that data from multiple subspaces can potentially be utilized. Taking the 8-bit S-box of SKINNY as an example, when the condition mask is $0x01$, if the condition set is $(0x01, 0)$, then $\mathbb{C}_{\mathcal{C}}(0x0a, 0x28) = -1$; while if the condition is $(0x01, 1)$, then $\mathbb{C}_{\mathcal{C}}(0x02, 0x28) = -1$. However, an effective method to fully exploit this information is non-trivial, and we leave this as an open problem for future work.

*More Advanced Key Recovery.* We employ the Fourier transform for key recovery. Recently, the Walsh spectrum puncturing technique has been used to reduce the complexity of the Fourier transform by nullifying certain insignificant spectrum points [15,16]. The effective integration of the Walsh spectrum puncturing technique with conditional linear cryptanalysis needs further investigation.

*Extension to Other Symmetric Primitives.* We mainly study the conditional correlations of S-boxes and focus on SPN-based ciphers in this paper. How to apply the method to ARX and AND-RX ciphers and propose a generalized tool similar to CLAT for nonlinear operations such as modular addition and AND operations is also interesting.

## 7 Conclusion

This paper presents a generalized framework for conditional linear cryptanalysis, which incorporates the proposed CLAT, capturing statistical linear properties under different subspaces; an automatic search model for identifying conditional linear trails; and procedures for distinguishing and key recovery attacks. We apply this framework to several AES-like ciphers, derive refined conditional linear security evaluations, and demonstrate improved linear attacks on them. Our framework indicates that the security bounds established by the wide-trail strategy cannot be directly used to evaluate resistance to conditional linear cryptanalysis. This highlights the need to take conditional data filtering into account when evaluating AES-like designs. Future work could focus on several directions, including more efficient data usage, more advanced key recovery techniques, and extensions to other symmetric primitives.

**Acknowledgments.** The work of this paper was supported by the National Natural Science Foundation of China under Grant No. 62372464.

## References

1. Ashur, T., Takke, E.: A new linear distinguisher for four-round AES. J. Cryptol. **38**(4), 30 (2025). https://doi.org/10.1007/S00145-025-09550-9
2. Banik, S., Bogdanov, A., Isobe, T., Shibutani, K., Hiwatari, H., Akishita, T., Regazzoni, F.: Midori: A block cipher for low energy. In: Iwata, T., Cheon, J.H. (eds.) Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9453, pp. 411–436. Springer (2015), https://doi.org/10.1007/978-3-662-48800-3_17
3. Bar-On, A., Dunkelman, O., Keller, N., Weizman, A.: DLCT: A new tool for differential-linear cryptanalysis. In: Ishai, Y., Rijmen, V. (eds.) Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11476, pp. 313–342. Springer (2019), https://doi.org/10.1007/978-3-030-17653-2_11

4. Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., Sim, S.M.: The SKINNY family of block ciphers and its low-latency variant MANTIS. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9815, pp. 123–153. Springer (2016), https://doi.org/10.1007/978-3-662-53008-5_5
5. Biham, E., Carmeli, Y.: An improvement of linear cryptanalysis with addition operations with applications to FEAL-8X. In: Joux, A., Youssef, A.M. (eds.) Selected Areas in Cryptography - SAC 2014 - 21st International Conference, Montreal, QC, Canada, August 14-15, 2014, Revised Selected Papers. Lecture Notes in Computer Science, vol. 8781, pp. 59–76. Springer (2014), https://doi.org/10.1007/978-3-319-13051-4_4
6. Biham, E., Perle, S.: Conditional linear cryptanalysis - cryptanalysis of DES with less than 242 complexity. IACR Trans. Symmetric Cryptol. **2018**(3), 215–264 (2018). https://doi.org/10.13154/TOSC.V2018.I3.215-264
7. Collard, B., Standaert, F., Quisquater, J.: Improving the time complexity of Matsui's linear cryptanalysis. In: Nam, K., Rhee, G. (eds.) Information Security and Cryptology - ICISC 2007, 10th International Conference, Seoul, Korea, November 29-30, 2007, Proceedings. Lecture Notes in Computer Science, vol. 4817, pp. 77–88. Springer (2007), https://doi.org/10.1007/978-3-540-76788-6_7
8. Daemen, J., Govaerts, R., Vandewalle, J.: Correlation matrices. In: Preneel, B. (ed.) Fast Software Encryption: Second International Workshop. Leuven, Belgium, 14-16 December 1994, Proceedings. Lecture Notes in Computer Science, vol. 1008, pp. 275–285. Springer (1994), https://doi.org/10.1007/3-540-60590-8_21
9. Daemen, J., Rijmen, V.: The block cipher Rijndael. In: Quisquater, J.J., Schneier, B. (eds.) Smart Card Research and Applications. pp. 277–284. Springer Berlin Heidelberg, Berlin, Heidelberg (2000)
10. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Information Security and Cryptography, Springer (2002), https://doi.org/10.1007/978-3-662-04722-4
11. Delaune, S., Derbez, P., Vavrille, M.: Catching the fastest boomerangs application to SKINNY. IACR Trans. Symmetric Cryptol. **2020**(4), 104–129 (2020), https://doi.org/10.46586/tosc.v2020.i4.104-129
12. 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
13. Flórez-Gutiérrez, A.: Optimising linear key recovery attacks with affine Walsh transform pruning. In: Agrawal, S., Lin, D. (eds.) Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part IV. Lecture Notes in Computer Science, vol. 13794, pp. 447–476. Springer (2022), https://doi.org/10.1007/978-3-031-22972-5_16
14. Flórez-Gutiérrez, A., Naya-Plasencia, M.: Improving key-recovery in linear attacks: Application to 28-round PRESENT. In: Canteaut, A., Ishai, Y. (eds.) Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12105, pp. 221–249. Springer (2020), https://doi.org/10.1007/978-3-030-45721-1_9

15. Flórez-Gutiérrez, A., Todo, Y.: Improving linear key recovery attacks using Walsh spectrum puncturing. In: Joye, M., Leander, G. (eds.) Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part I. Lecture Notes in Computer Science, vol. 14651, pp. 187–216. Springer (2024), https://doi.org/10.1007/978-3-031-58716-0_7
16. Flórez-Gutiérrez, A., Todo, Y.: Improving linear key recovery attacks using Walsh spectrum puncturing. J. Cryptol. **38**(4), 34 (2025), https://doi.org/10.1007/s00145-025-09554-5
17. Fu, K., Wang, M., Guo, Y., Sun, S., Hu, L.: MILP-based automatic search algorithms for differential and linear trails for Speck. In: Peyrin, T. (ed.) Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers. pp. 268–288. Lecture Notes in Computer Science, Springer (2016), https://doi.org/10.1007/978-3-662-52993-5_14
18. Guo, J., Peyrin, T., Poschmann, A., Robshaw, M.J.B.: The LED block cipher. In: Preneel, B., Takagi, T. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6917, pp. 326–341. Springer (2011), https://doi.org/10.1007/978-3-642-23951-9_22
19. 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
20. 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
21. Jr., B.S.K., Robshaw, M.J.B.: Linear cryptanalysis using multiple approximations. In: Desmedt, Y. (ed.) Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings. Lecture Notes in Computer Science, vol. 839, pp. 26–39. Springer (1994), https://doi.org/10.1007/3-540-48658-5_4
22. Knudsen, L.R., Rijmen, V.: Known-key distinguishers for some block ciphers. In: Kurosawa, K. (ed.) Advances in Cryptology - ASIACRYPT 2007, 13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, December 2-6, 2007, Proceedings. Lecture Notes in Computer Science, vol. 4833, pp. 315–324. Springer (2007), https://doi.org/10.1007/978-3-540-76900-2_19
23. Kwon, D., Kim, J., Park, S., Sung, S.H., Sohn, Y., Song, J.H., Yeom, Y., Yoon, E., Lee, S., Lee, J., Chee, S., Han, D., Hong, J.: New block cipher: ARIA. In: Lim, J.I., Lee, D.H. (eds.) Information Security and Cryptology - ICISC 2003, 6th International Conference, Seoul, Korea, November 27-28, 2003, Revised Papers. Lecture Notes in Computer Science, vol. 2971, pp. 432–445. Springer (2003), https://doi.org/10.1007/978-3-540-24691-6_32
24. Leander, G.: On linear hulls, statistical saturation attacks, PRESENT and a cryptanalysis of PUFFIN. In: Paterson, K.G. (ed.) Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6632, pp. 303–322. Springer (2011), https://doi.org/10.1007/978-3-642-20465-4_18
25. Li, Y., Wang, M., Ou, H., Wang, S.: Improved integral analysis on lightweight block cipher Midori. In: 2019 IEEE 5th International Conference on Computer

and Communications (ICCC). pp. 1494–1498 (2019). https://doi.org/10.1109/ICCC47050.2019.9064222
26. Li, Y., Wu, W., Zhang, L.: Integral attacks on reduced-round ARIA block cipher. In: Kwak, J., Deng, R.H., Won, Y., Wang, G. (eds.) Information Security, Practice and Experience, 6th International Conference, ISPEC 2010, Seoul, Korea, May 12-13, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6047, pp. 19–29. Springer (2010), https://doi.org/10.1007/978-3-642-12827-1_2
27. Matsui, M.: Linear cryptanalysis method for DES cipher. In: Helleseth, T. (ed.) Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings. Lecture Notes in Computer Science, vol. 765, pp. 386–397. Springer (1993), https://doi.org/10.1007/3-540-48285-7_33
28. Matsui, M.: The first experimental cryptanalysis of the data encryption standard. In: Desmedt, Y. (ed.) Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings. Lecture Notes in Computer Science, vol. 839, pp. 1–11. Springer (1994), https://doi.org/10.1007/3-540-48658-5_1
29. Mouha, N., Wang, Q., Gu, D., Preneel, B.: Differential and linear cryptanalysis using mixed-integer linear programming. In: Wu, C., Yung, M., Lin, D. (eds.) Information Security and Cryptology - 7th International Conference, Inscrypt 2011, Beijing, China, November 30 - December 3, 2011. Revised Selected Papers. Lecture Notes in Computer Science, vol. 7537, pp. 57–76. Springer (2011), https://doi.org/10.1007/978-3-642-34704-7_5
30. Newman, D.J.: The Double Dixie Cup Problem. American Mathematical Monthly **67**(1), 58 (Jan 1960). https://doi.org/10.2307/2308930
31. Sajadieh, M., Mirzaei, A., Mala, H., Rijmen, V.: A new counting method to bound the number of active S-boxes in Rijndael and 3D. Des. Codes Cryptogr. **83**(2), 327–343 (2017), https://doi.org/10.1007/s10623-016-0217-4
32. Selçuk, A.A.: On probability of success in linear and differential cryptanalysis. J. Cryptol. **21**(1), 131–147 (2008), https://doi.org/10.1007/s00145-007-9013-7
33. Sun, S., Hu, L., Wang, P., Qiao, K., Ma, X., Song, L.: Automatic security evaluation and (related-key) differential characteristic search: Application to SIMON, PRESENT, LBlock, DES(L) and other bit-oriented block ciphers. In: Sarkar, P., Iwata, T. (eds.) Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I. Lecture Notes in Computer Science, vol. 8873, pp. 158–178. Springer (2014), https://doi.org/10.1007/978-3-662-45611-8_9
34. Taka, K., Sakamoto, K., Ito, R., Shiba, R., Utsumi, S., Isobe, T.: Divide-and-conquer SAT for exploring optimal differential and linear characteristics and its applications. IACR Trans. Symmetric Cryptol. **2025**(3), 516–576 (2025), https://doi.org/10.46586/tosc.v2025.i3.516-576

## A Proof of Theorem 1

**Theorem 1.** *Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^p$, $G : \mathbb{F}_2^p \to \mathbb{F}_2^q$, and $H : \mathbb{F}_2^q \to \mathbb{F}_2^m$ be three vectorial Boolean functions. Let $X : \mathbb{F}_2^n \to \mathbb{F}_2^r$ and $Y : \mathbb{F}_2^m \to \mathbb{F}_2^s$ also be two vectorial Boolean functions. For $a \in \mathbb{F}_2^r$ and $b \in \mathbb{F}_2^s$, denote*

$$\mathcal{P}_a = \{x \in \mathbb{F}_2^n : X(x) = a\} \text{ and } \mathcal{Q}_b = \{z \in \mathbb{F}_2^q : Y(H(z)) = b\}.$$

*Moreover, denote the set of data satisfying the two condition sets by*

$$\mathcal{S}_{a,b} = \{x \in \mathcal{P}_a : G(F(x)) \in \mathcal{Q}_b\}.$$

*Assume that $\mathcal{P}_a$, $\mathcal{Q}_b$, and $\mathcal{S}_{a,b}$ are nonempty. Then, for any input mask $u \in \mathbb{F}_2^n$ and output mask $v \in \mathbb{F}_2^m$, we have*

$$\mathbb{C}^{H\circ G\circ F}\big|_{\mathcal{S}_{a,b}}(u,v) = \frac{|\mathcal{P}_a| \cdot |\mathcal{Q}_b|}{2^q \cdot |\mathcal{S}_{a,b}|} \sum_{\alpha\in\mathbb{F}_2^p, \beta\in\mathbb{F}_2^q} \mathbb{C}^H\big|_{\mathcal{Q}_b}(\beta,v) \cdot \mathbb{C}^G(\alpha,\beta) \cdot \mathbb{C}^F\big|_{\mathcal{P}_a}(u,\alpha).$$

*Proof.* Let us first recall that for any pair of masks $u \in \mathbb{F}_2^n$ and $v \in \mathbb{F}_2^m$, the correlation of the approximation $(u, v)$ with any vectorial Boolean function $K : \mathbb{F}_2^n \to \mathbb{F}_2^m$ can be computed as

$$\mathbb{C}^K(u,v) = \frac{1}{|\mathbb{F}_2^n|} \sum_{x\in\mathbb{F}_2^n} (-1)^{u\cdot x \oplus v\cdot K(x)},$$

while conditional correlation on the data subset $\mathcal{S}$ can be computed as

$$\mathbb{C}^K(u,v)\big|_{\mathcal{S}} = \frac{1}{|\mathcal{S}|} \sum_{x\in\mathcal{S}} (-1)^{u\cdot x \oplus v\cdot K(x)}.$$

Accordingly, for any $u \in \mathbb{F}_2^n$ and $v \in \mathbb{F}_2^m$, it holds that

$$\begin{aligned}
&\sum_{\beta\in\mathbb{F}_2^q} \sum_{\alpha\in\mathbb{F}_2^p} \left(\mathbb{C}^H\big|_{\mathcal{Q}_b}(\beta,v)\right) \cdot \mathbb{C}^G(\alpha,\beta) \cdot \left(\mathbb{C}^F\big|_{\mathcal{P}_a}(u,\alpha)\right) \\
&= \sum_{\beta,\alpha} \left(\frac{1}{|\mathcal{Q}_b|} \sum_{z\in\mathcal{Q}_b} (-1)^{\beta\cdot z \oplus v\cdot H(z)}\right) \cdot \left(\frac{1}{2^p} \sum_{y\in\mathbb{F}_2^p} (-1)^{\alpha\cdot y \oplus \beta\cdot G(y)}\right) \\
&\qquad \cdot \left(\frac{1}{|\mathcal{P}_a|} \sum_{x\in\mathcal{P}_a} (-1)^{u\cdot x \oplus \alpha\cdot F(x)}\right) \\
&= \frac{1}{2^p\,|\mathcal{P}_a|\,|\mathcal{Q}_b|} \sum_{z\in\mathcal{Q}_b} \sum_{y\in\mathbb{F}_2^p} \sum_{x\in\mathcal{P}_a} (-1)^{u\cdot x \oplus v\cdot H(z)} \cdot \sum_{\alpha\in\mathbb{F}_2^p} (-1)^{\alpha\cdot(y\oplus F(x))} \\
&\qquad \cdot \sum_{\beta\in\mathbb{F}_2^q} (-1)^{\beta\cdot(z\oplus G(y))}.
\end{aligned}$$

By the orthogonality of characters over binary vector spaces, we have

$$\sum_{\alpha\in\mathbb{F}_2^p}(-1)^{\alpha\cdot(y\oplus F(x))}=2^p\delta_{y,F(x)}$$

and

$$\sum_{\beta\in\mathbb{F}_2^q}(-1)^{\beta\cdot(z\oplus G(y))}=2^q\delta_{z,G(y)}.$$

It immediately follows that

$$\begin{aligned}
&\sum_{\beta\in\mathbb{F}_2^q}\sum_{\alpha\in\mathbb{F}_2^p}\left(\mathbb{C}^H\big|_{\mathcal{Q}_b}(\beta,v)\right)\cdot\mathbb{C}^G(\alpha,\beta)\cdot\left(\mathbb{C}^F\big|_{\mathcal{P}_a}(u,\alpha)\right)\\
&=\frac{2^q}{|\mathcal{P}_a|\,|\mathcal{Q}_b|}\sum_{x\in\mathcal{S}_{a,b}}(-1)^{u\cdot x\oplus v\cdot H(G(F(x)))}\\
&=\frac{2^q\cdot|\mathcal{S}_{a,b}|}{|\mathcal{P}_a|\cdot|\mathcal{Q}_b|}\left(C^{H\circ G\circ F}\big|_{\mathcal{S}_{a,b}}(u,v)\right).
\end{aligned}$$

□

## B Details of the Experimental Verification of the FWHT Technique

To verify the correctness of the FWHT-based key recovery, we perform a simple experiment on 3-round Midori-128. The attack diagram is provided in Figure 3. We use a 2-round conditional linear trail with a conditional correlation of $2^{-3}$ under the condition set $\mathcal{C} = \{(0x50, 1)\}$. An additional round is appended before the linear trail to recover the key bits. There are in total 24 key bits that need to be guessed, and the violet squares in the first round indicate the bytes known after the key guess. According to Formulas (6) and (7), when the data complexity reaches $2^{12.95}$, the success probability of recovering the correct key bits exceeds 99%. We perform a number of independent experiments, each using $2^{13}$ samples to perform the practical key recovery attack. The empirical correlations under the correct key fluctuate around $2^{-3}$ and always rank the highest. This demonstrates the validity of our key recovery procedure.

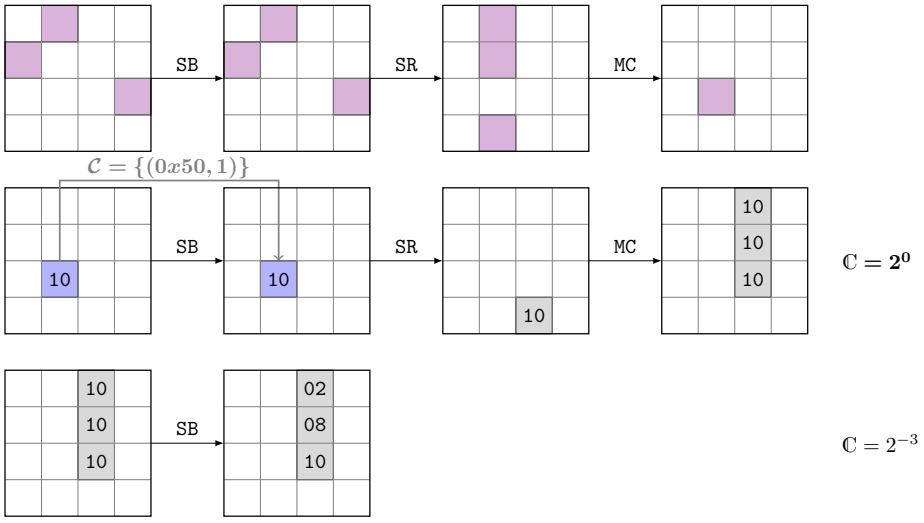



**Fig. 3.** Key recovery attack against 3-round Midori-128.

## C Experimental Details for Conditional Linear Correlations

In this appendix, we provide the detailed experimental settings for the conditional linear correlations reported in Section 5.2.

*AES.* For 3-round AES, we identify a conditional linear trail with a conditional linear weight of 36.64, involving 24 conditions and exhibiting a conditional correlation of $2^{-6.32}$. We conduct 64 independent experiments, each using $2^{40}$ known plaintext-ciphertext pairs to evaluate the empirical conditional correlations. The number of plaintext-ciphertext pairs in each experiment that satisfy the specified conditions ranges from $2^{15.990}$ to $2^{16.016}$, which closely matches the expected fraction of $2^{-24}$ of the entire data set. Furthermore, the empirical conditional correlations computed from the filtered pairs range from $2^{-7.257}$ to $2^{-5.579}$, with a mean of $2^{-6.323}$ and a standard deviation of $2^{-8.102}$.

*LED.* For 4-round LED, we identify a conditional linear trail with a conditional linear weight of 30, involving 20 conditions and exhibiting a conditional correlation of $2^{-5}$. We conduct 64 independent experiments, each using $2^{34}$ known plaintext-ciphertext pairs. After filtering by the condition set, the number of remaining data ranges from $2^{13.985}$ to $2^{14.026}$. The empirical conditional correlations computed from the filtered data range from $2^{-6.154}$ to $2^{-4.287}$, with a mean of $2^{-4.982}$ and a standard deviation of $2^{-7.053}$. These empirical correlations are distributed around the theoretical value $2^{-5}$.

*Midori.* For 5-round Midori, we identify a conditional linear trail with a conditional linear weight of 32, involving 18 conditions and exhibiting a conditional correlation of $2^{-7}$. We perform 64 independent experiments with $2^{37}$ known plaintext-ciphertext pairs in each experiment. The number of pairs satisfying the specified conditions ranges from $2^{18.995}$ to $2^{19.005}$, which agrees with the expected filtering ratio $2^{-18}$. The empirical conditional correlations obtained from the filtered samples are within the range $2^{-7.657}$ to $2^{-6.460}$, with a mean of $2^{-7.03}$ and a standard deviation of $2^{-9.572}$. The empirical correlations are close to the theoretical value $2^{-7}$.

*SKINNY.* For 6-round SKINNY, we identify a conditional linear trail with correlation $2^{-8}$ under 11 conditions. Since SKINNY exhibits a strong linear hull effect, the actual correlation cannot be accurately captured by this single trail alone. To evaluate the behavior of the identified approximation, we conduct 128 independent experiments, each using $2^{31}$ known plaintext-ciphertext pairs. After filtering by the prescribed condition set, the number of remaining pairs in each experiment ranges from $2^{19.996}$ to $2^{20.003}$, which matches the expected filtering ratio of $2^{-11}$. Among these experiments, 76 yield empirical correlations greater than $2^{-8.5}$, which is consistent with Theorem 2 in [24].

## D The Conditional Linear Trail of 4-Round AES

Figure 4 shows the conditional linear trail of 4-round AES and each conditional active S-box involves three conditions. The blue squares represent active bytes in the conditional rounds and the gray squares represent active bytes in the standard linear trail. The input condition function $X_{in}$ and the output condition function $X_{out}$ are defined as below. When the outputs of both functions are zero vectors, the linear trail and its correlation, as illustrated in Figure 4, are satisfied.

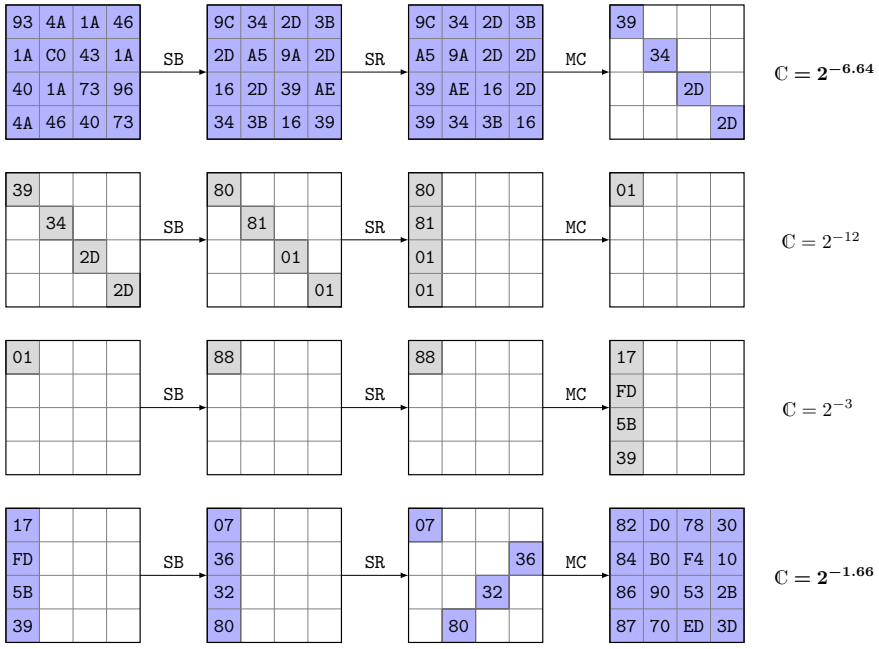



**Fig. 4.** The conditional linear trail of 4-round AES.

$$
\begin{aligned}
&X_{in} : \mathbb{F}_2^{128} \to \mathbb{F}_2^{48} : \boldsymbol{x} \mapsto \\
&\texttt{(20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00}^{\top}\boldsymbol{x}, \\
&\texttt{ 2A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00}^{\top}\boldsymbol{x}, \\
&\texttt{ 56 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00}^{\top}\boldsymbol{x}, \\
&\texttt{ 00 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00}^{\top}\boldsymbol{x}, \\
&\texttt{ 00 76 00 00 00 00 00 00 00 00 00 00 00 00 00 00}^{\top}\boldsymbol{x}, \\
&\texttt{ 00 C5 00 00 00 00 00 00 00 00 00 00 00 00 00 00}^{\top}\boldsymbol{x}, \\
&\texttt{ 00 00 11 00 00 00 00 00 00 00 00 00 00 00 00 00}^{\top}\boldsymbol{x}, \\
&\texttt{ 00 00 16 00 00 00 00 00 00 00 00 00 00 00 00 00}^{\top}\boldsymbol{x}, \\
&\texttt{ 00 00 33 00 00 00 00 00 00 00 00 00 00 00 00 00}^{\top}\boldsymbol{x}, \\
&\texttt{ 00 00 00 07 00 00 00 00 00 00 00 00 00 00 00 00}^{\top}\boldsymbol{x},
\end{aligned}
$$

$$
\begin{array}{l}
\texttt{00 00 00 20 00 00 00 00 00 00 00 00 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 37 00 00 00 00 00 00 00 00 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 07 00 00 00 00 00 00 00 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 20 00 00 00 00 00 00 00 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 37 00 00 00 00 00 00 00 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 08 00 00 00 00 00 00 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 0B 00 00 00 00 00 00 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 34 00 00 00 00 00 00 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 20 00 00 00 00 00 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 76 00 00 00 00 00 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 C5 00 00 00 00 00 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 0A 00 00 00 00 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 17 00 00 00 00 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 22 00 00 00 00 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 20 00 00 00 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 76 00 00 00 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 C5 00 00 00 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 00 04 00 00 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 00 11 00 00 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 00 C2 00 00 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 00 00 07 00 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 00 00 09 00 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 00 00 C6 00 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 00 00 00 11 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 00 00 00 16 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 00 00 00 33 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 00 00 00 00 0A 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 00 00 00 00 17 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 00 00 00 00 22 00 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 00 00 00 00 00 20 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 00 00 00 00 00 76 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 00 00 00 00 00 C5 00 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 00 00 00 00 00 00 0B 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 00 00 00 00 00 00 30 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 00 00 00 00 00 00 4F 00}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 07}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 09}^\top \boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 C6}^\top \boldsymbol{x})
\end{array}
$$

$$
\begin{array}{l}
X_{out} : \mathbb{F}_2^{128} \to \mathbb{F}_2^{12} : \boldsymbol{x} \mapsto \\
(\texttt{05 08 0C 0F 00 00 00 00 00 00 00 00 00 00 00 00}^\top \boldsymbol{x}, \\
\texttt{2E FA B6 72 00 00 00 00 00 00 00 00 00 00 00 00}^\top \boldsymbol{x},
\end{array}
$$

$$
\begin{array}{l}
\texttt{7D 17 CE 86 00 00 00 00 00 00 00 00 00 00 00 00}^{\top}\boldsymbol{x}, \\
\texttt{00 00 00 00 CE 49 CB 46 00 00 00 00 00 00 00 00}^{\top}\boldsymbol{x}, \\
\texttt{00 00 00 00 B6 AF A7 9D 00 00 00 00 00 00 00 00}^{\top}\boldsymbol{x}, \\
\texttt{00 00 00 00 12 6E BA 96 00 00 00 00 00 00 00 00}^{\top}\boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 28 EA A6 6F 00 00 00 00}^{\top}\boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 D8 C8 B8 E8 00 00 00 00}^{\top}\boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 C9 AE 9A 60 00 00 00 00}^{\top}\boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 00 00 00 00 CB 46 CE 49}^{\top}\boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 00 00 00 00 A7 9D B6 AF}^{\top}\boldsymbol{x}, \\
\texttt{00 00 00 00 00 00 00 00 00 00 00 00 BA 96 12 6E}^{\top}\boldsymbol{x})
\end{array}
$$

## E Key Recovery Attack on 7-Round Rijndael-256

Figure 5 shows the key recovery process for 7-round Rijndael-256. We employ a 5-round conditional linear trail, where the blue squares represent active bytes in the conditional rounds and the gray squares represent the active bytes in the standard linear trail. We append one round before and after the linear trail, respectively, and the violet squares indicate the known bytes obtained after guessing the key.

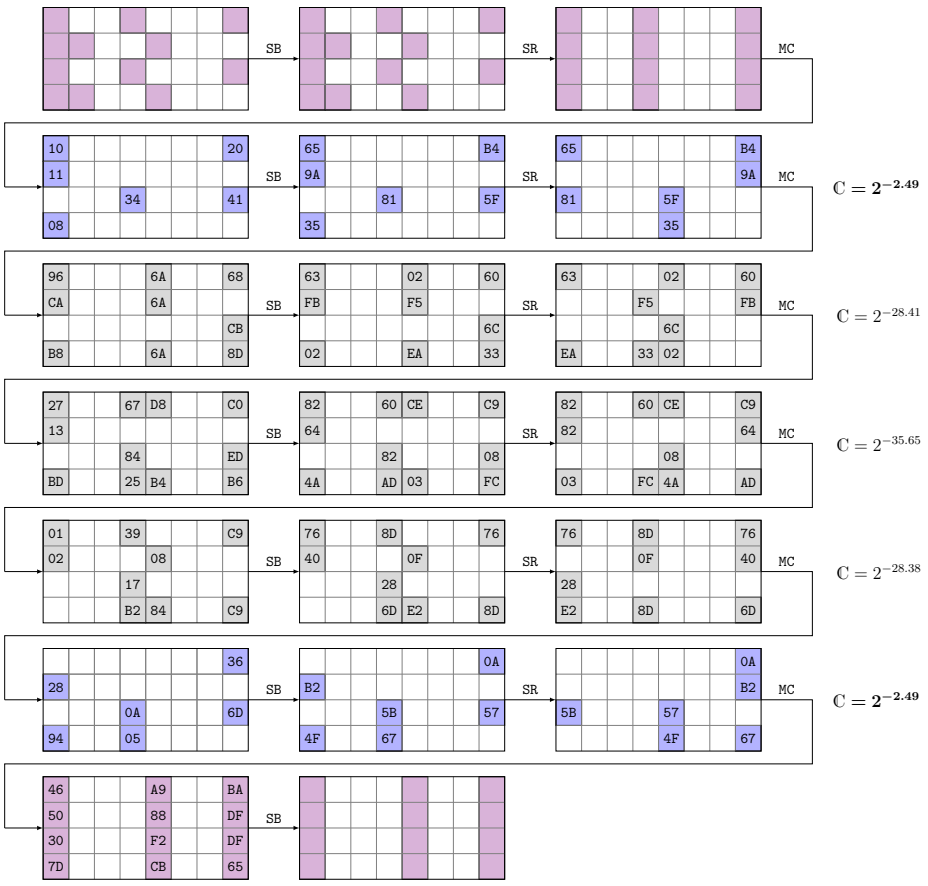



**Fig. 5.** Key recovery attack against 7-round Rijndael-256.

Each conditional active S-box in Figure 5 involves three conditions. The input condition function $X_{in}$ and the output condition function $X_{out}$ are defined as below. When the outputs of both functions are zero vectors, the conditional linear trail and its correlation, as shown in Figure 5, are satisfied.

$X_{in} : \mathbb{F}_2^{256} \to \mathbb{F}_2^{18} : \boldsymbol{x} \mapsto$

$$
\begin{pmatrix}
\texttt{3B}\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00 \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00^\top \boldsymbol{x}, \\
\texttt{54}\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00 \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00^\top \boldsymbol{x}, \\
\texttt{B6}\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00 \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00^\top \boldsymbol{x}, \\
00\ \texttt{1B}\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00 \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00^\top \boldsymbol{x}, \\
00\ \texttt{49}\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00 \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00^\top \boldsymbol{x}, \\
00\ \texttt{94}\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00 \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00^\top \boldsymbol{x}, \\
00\ 00\ 00\ \texttt{05}\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00 \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00^\top \boldsymbol{x}, \\
00\ 00\ 00\ \texttt{21}\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00 \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00^\top \boldsymbol{x}, \\
00\ 00\ 00\ \texttt{CA}\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00 \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00^\top \boldsymbol{x}, \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ \texttt{03}\ 00 \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00^\top \boldsymbol{x}, \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ \texttt{5C}\ 00 \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00^\top \boldsymbol{x}, \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ \texttt{9F}\ 00 \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00^\top \boldsymbol{x}, \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00 \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ \texttt{1A}\ 00\ 00\ 00^\top \boldsymbol{x}, \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00 \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ \texttt{4C}\ 00\ 00\ 00^\top \boldsymbol{x}, \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00 \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ \texttt{89}\ 00\ 00\ 00^\top \boldsymbol{x}, \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00 \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ \texttt{18}\ 00^\top \boldsymbol{x}, \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00 \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ \texttt{1C}\ 00^\top \boldsymbol{x}, \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00 \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ \texttt{A5}\ 00^\top \boldsymbol{x}
\end{pmatrix}
$$

$X_{out} : \mathbb{F}_2^{256} \to \mathbb{F}_2^{18} : \boldsymbol{x} \mapsto$

$$
\begin{pmatrix}
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00 \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ \texttt{EB}\ \texttt{59}\ \texttt{FC}\ 60^\top \boldsymbol{x}, \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00 \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ \texttt{2D}\ \texttt{1B}\ \texttt{8F}\ \texttt{FC}^\top \boldsymbol{x}, \\
00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00
\end{pmatrix}
$$

00 00 00 00 00 00 00 00 00 00 00 00 26 1D EC 43$^\top \boldsymbol{x}$,
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
78 E6 6C DB 00 00 00 00 00 00 00 00 00 00 00 00$^\top \boldsymbol{x}$,
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
E0 D4 C7 BD 00 00 00 00 00 00 00 00 00 00 00 00$^\top \boldsymbol{x}$,
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
45 2E 08 F8 00 00 00 00 00 00 00 00 00 00 00 00$^\top \boldsymbol{x}$,
E7 A6 62 25 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00$^\top \boldsymbol{x}$,
3C FA A9 76 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00$^\top \boldsymbol{x}$,
EF 15 F3 E1 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00$^\top \boldsymbol{x}$,
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 C6 45 C4 43$^\top \boldsymbol{x}$,
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 C0 93 5D 34$^\top \boldsymbol{x}$,
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 F1 CC D8 B7$^\top \boldsymbol{x}$,
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 3E D1 8B 42$^\top \boldsymbol{x}$,
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 7C A2 17 84$^\top \boldsymbol{x}$,
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 1D EC 43 26$^\top \boldsymbol{x}$,
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
24 E5 A3 67 00 00 00 00 00 00 00 00 00 00 00 00$^\top \boldsymbol{x}$,
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
91 03 FE 26 00 00 00 00 00 00 00 00 00 00 00 00$^\top \boldsymbol{x}$,
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1F 37 ED 66 00 00 00 00 00 00 00 00 00 00 00 00$^\top \boldsymbol{x}$)

## F Key Recovery Attack on 12-Round Midori-128

Figure 6 shows the key recovery process for 12-round Midori-128. We employ a 10-round conditional linear trail with blue squares representing the active bytes in the conditional rounds. The eight omitted rounds of linear trails are detailed in [34]. We append one round before and after the linear trail, respectively, and the violet squares indicate the known bytes obtained after guessing the key.

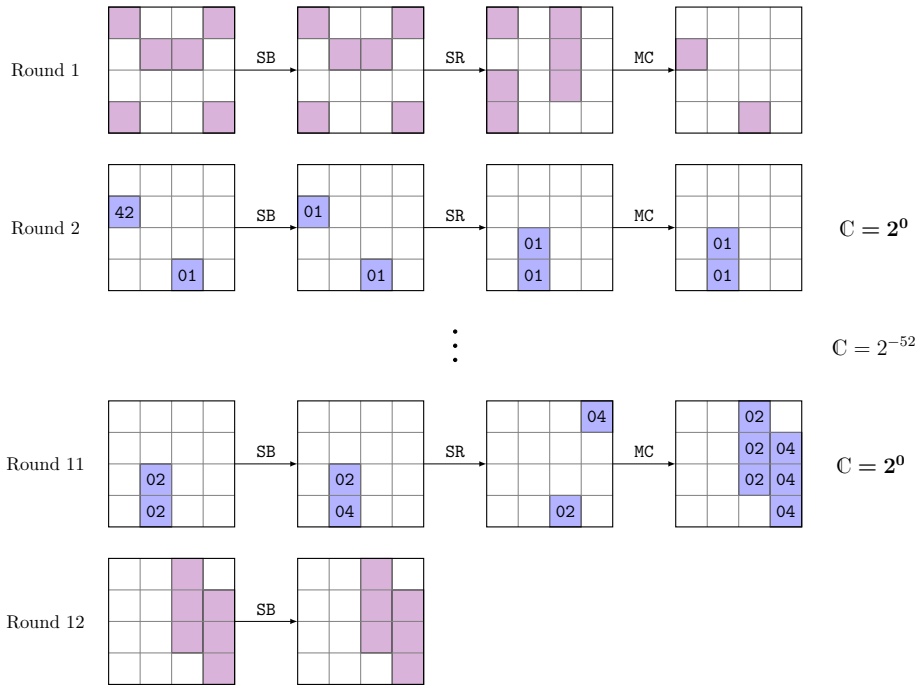



**Fig. 6.** Key recovery attack against 12-round Midori-128.

The input condition function $X_{in}$ and the output condition function $X_{out}$ are defined as below. When the output of $X_{in}$ is $(0, 0)$ and the output of $X_{out}$ is (0,1), the linear trail and its correlation, as illustrated in Figure 6, are satisfied.

$X_{in} : \mathbb{F}_2^{128} \to \mathbb{F}_2^2 : \boldsymbol{x} \mapsto$
$(00\ C2\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00^\top \boldsymbol{x},$
$00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 08\ 00\ 00\ 00\ 00^\top \boldsymbol{x})$

$X_{out} : \mathbb{F}_2^{128} \to \mathbb{F}_2^2 : \boldsymbol{x} \mapsto$
$(00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 01\ 01\ 01\ 00\ 00\ 00\ 00\ 00^\top \boldsymbol{x},$
$00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 84\ 84\ 84^\top \boldsymbol{x})$