IACR Transactions on Symmetric Cryptology
ISSN XXXX-XXXX, Vol. 0, No. 0, pp. 1–31. DOI:XXXXXXXX

Differential Pattern Transition

Characterizing the Differential Behavior of AES-like Linear Layers

Jianhua Wang[1,2,3], Tao Huang[2(✉)], Siwei Sun[1,3], Hailun Yan[1(✉)], Guang Zeng[2] and Shuang Wu[2]

[1] School of Cryptology, University of Chinese Academy of Sciences, Beijing, China
wangjianhua@ucas.ac.cn,sunsiwei@ucas.ac.cn,hailun.yan@ucas.ac.cn
[2] Shield Lab, Huawei Technologies Co., Ltd.
huangtao80@huawei.com,zengguang13@huawei.com,Wu.Shuang@huawei.com
[3] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing, China

Abstract. This paper introduces a new cryptographic notion for diffusion matrices, termed the *Differential Pattern Transition* (DPT). Building on this notion, we develop a systematic framework for describing the differential behavior of diffusion layers over multiple rounds in AES-like block ciphers. Specifically, the DPT framework enables a finer-grained evaluation of diffusion strength against differential attacks, allowing distinctions even among matrices sharing the same branch number. Furthermore, the DPT framework facilitates the classification of shuffle layers and assists in identifying permutation layers that maximize differential resistance.
As a case study, we apply the DPT framework to the diffusion matrices used in MIDORI, PRINCE, QARMA, and AES, as well as a lightweight MDS matrix proposed in [SS16]. The results show that DPT provides both theoretical insights and practical guidance for the selection and design of diffusion and shuffle layers in secure and efficient block cipher constructions.

Keywords: Differential pattern transition · Diffusion matrix · Shuffle layer · AES-like cipher

1 Introduction

In the realm of symmetric cryptographic design, the substitution-permutation network (SPN) [Sha49] stands out as a prominent design paradigm. At the core of an SPN is its layered architecture, consisting of substitution layers (non-linear transformations) and linear layers. This construction is widely adopted in symmetric cipher design, with the Advanced Encryption Standard (AES) [DR02] being a notable example. Moreover, numerous SPN-based lightweight and low-latency block ciphers have been proposed, including LED [GPPR11], PRINCE [BCG+12], MIDORI [BBI+15], and SKINNY [BJK+16].

In the SPN cipher, the linear layer plays a significant role in shaping the cryptographic strength. It is a crucial component for diffusing the non-linear properties introduced by the substitution layer across the cipher state. This diffusion effectively mitigates the risk of cryptanalytic attacks by ensuring that small alterations in the input lead to widespread changes in the output. A well-designed linear layer can significantly increase the cipher's resistance to various attacks, such as differential cryptanalysis [BS91], linear cryptanalysis [Mat94], integral attacks [DKR97, KW02], among others, thereby enhancing the overall security of the cipher.

To construct a linear layer, two commonly used components are shuffle transformations and diffusion matrices. The shuffle operation rearranges the bits in the cipher state, helping to scramble the correspondence between the input and output positions, which

Licensed under Creative Commons License CC-BY 4.0.

increases resistance to structural cryptanalysis. Diffusion matrices, on the other hand, spread each input bit into multiple output bits through a linear mapping, typically realized as a matrix multiplication over a finite field. This mapping ensures that a small change in the input propagates to a significant portion of the output, thereby creating strong diffusion and improving resistance to attacks. The diffusion capability of such matrices is often characterized by the branch number [Dae95, DR02, LS16, ZWFS09], which provides a classical metric for measuring their effectiveness. This metric has played an important role in many SPN cipher designs.

The notion of differential/linear branch number was introduced in [Dae95, DR02]. It is defined as the minimum of the sum of the Hamming weights of a non-zero input difference (or output mask) and its corresponding output difference (or input mask) under the matrix. Numerically, it provides a lower bound on the number of active S-boxes in differential and linear trails over two rounds. Moreover, for `AES`-like ciphers, if the shuffle layer satisfies the wide-trail strategy—that is, mapping cells of the same column into different columns—then the lower bound on the number of active S-boxes over four rounds is given by the square of the branch number.

This principle has guided the design of many block ciphers, including `LED` [GPPR11], `PRINCE` [BCG+12], `mCrypton` [LK06], and `PHOTON` [GPP11], among others. Building on this idea, researchers have also explored the design of diffusion matrices with larger branch numbers. Notable directions include the construction of lightweight MDS or almost MDS matrices [SKOP15, SS16, LW17, LSL+19, LS16, LSS+19, YZW21], as well as the development of large lightweight diffusion matrices [LRL+24, EHDRM23].

However, the branch number alone does not fully capture the impact of diffusion matrices on the differential or linear resistance of a cipher over multiple rounds, especially beyond four rounds. This constitutes a critical gap in our understanding, as multi-round operations are standard practice in block cipher design to amplify security. In particular, ciphers with diffusion matrices sharing the same branch number may exhibit significantly different levels of resistance when extended to many rounds. A deeper investigation into how diffusion matrices influence multi-round behaviors is therefore essential, as it can guide the development of more effective design strategies that not only optimize lightweight or low-latency implementation costs but also enhance security of block ciphers.

Our Contributions

In this work, we address the above gap by introducing a new cryptographic notion for diffusion matrices, termed the *Differential Pattern Transition* (DPT). Building upon this notion, we develop a systematic framework for evaluating how differential patterns propagate through diffusion layer across multiple rounds in `AES`-like ciphers, thereby extending beyond the traditional two-round guarantees provided by the branch number. Our main contributions are summarized as follows:

- **Formalization of the Differential Pattern Transition.** We formally introduce the DPT notion and develop its theoretical foundation, comprising a partial order relation between two DPTs and equivalence relations for permutation pairs applied on both sides of a diffusion matrix. This foundation provides a principled basis for analyzing linear layers.

- **Systematic framework for multi-round analysis.** Leveraging DPT, we capture the propagation of differential patterns across multiple rounds in `AES`-like ciphers. The framework enables fine-grained comparison of diffusion matrices, even when they share the same branch number, and supports the classification and filtering of shuffle layers. A branch-and-bound strategy further accelerates the identification of optimal shuffle-layer configurations.

- **Application to existing cipher designs.** We apply DPT framework to the diffusion matrices of lightweight, low-latency ciphers (MIDORI, PRINCE, QARMA, and the lightweight matrix from [SS16]) as well as on the diffusion matrix of the standard cipher AES. Our analysis demonstrates that DPT can detect subtle differences in differential resistance and provide actionable guidance for designing more efficient linear layers.
- **Extension to heterogeneous diffusion matrices.** We extend DPT framework to ciphers using different diffusion matrices in odd and even rounds. The results show that alternating matrices can enhance differential resistance while reducing latency or area, highlighting DPT's utility in exploring heterogeneous-round design strategies for low-latency and lightweight AES-like ciphers.

Outline. This paper is organized as follows. Section 2 reviews some related background knowledge. In Section 3, we introduce the notion of DPT and establish its theoretical foundation. Section 4 presents the DPT framework to AES-like ciphers. As the applications, we study several diffusion matrices used in typical ciphers in Section 5. Finally, Section 6 summarizes the paper.

2 Preliminaries

2.1 AES-like Ciphers

A substitution–permutation network (SPN) is a widely used structure in block ciphers, adopted in designs such as AES, PRESENT, and LED. In this paper, we focus on the AES-like SPN, where the internal state is represented as an $r \times c$ cell matrix. Each round consists of four operations: the S-box layer (S), the cell shuffle layer (P), the MixColumns operation (MC), and the subkey addition (AK). The overall process can be expressed as

$$\mathtt{S} \rightarrow \mathtt{P} \rightarrow \mathtt{MC} \rightarrow \mathtt{AK} \rightarrow \mathtt{S} \rightarrow \mathtt{P} \rightarrow \mathtt{MC} \rightarrow \mathtt{AK} \rightarrow \cdots \tag{1}$$

2.2 Differential Characteristics and Linear Characteristics

The differential probability between an input difference Δ_{in} and an output difference Δ_{out} for a function F is defined as

$$DP_F[\Delta_{in}, \Delta_{out}] = \frac{\#\{x \in \{0,1\}^n \mid F(x) \oplus F(x \oplus \Delta_{in}) = \Delta_{out}\}}{2^n}.$$

A differential characteristic of the cipher $E_k = F_r \circ \cdots \circ F_1$ is denoted by $(\Delta_0, \Delta_1, \ldots, \Delta_r)$, where Δ_{i-1} and Δ_i represent the input and output differences of the function F_i, respectively. Assuming that the functions F_i are independent, the probability of this characteristic is given by

$$\prod_{i=1}^{r} DP_{F_i}[\Delta_{i-1}, \Delta_i].$$

The linear correlation between an input mask Γ_{in} and an output mask Γ_{out} for a function F is defined as

$$LC_F[\Gamma_{in}, \Gamma_{out}] = \frac{\#\{x \in \{0,1\}^n \mid \Gamma_{in} \cdot x \oplus \Gamma_{out} \cdot F(x) = 0\}}{2^{n-1}} - 1.$$

A linear characteristic of the cipher $E_k = F_r \circ \cdots \circ F_1$ is denoted by $(\Gamma_0, \Gamma_1, \ldots, \Gamma_r)$, and its corresponding correlation is defined as

$$\prod_{i=1}^{r} LC_{F_i}[\Gamma_{i-1}, \Gamma_i],$$

under the assumption that the functions F_i are independent.

In ciphers that include an S-box layer, an S-box is said to be differentially active (resp. linearly active) if its input difference (resp. output mask) is nonzero. Typically, the maximum differential probability (resp. linear correlation) of a cipher is estimated by the product of the lower bound on the number of active S-boxes and the maximum differential probability (resp. maximum correlation) of a single S-box. Therefore, deriving a tight lower bound on the number of active S-boxes serves as an essential means of evaluating the cipher's resistance against differential and linear cryptanalysis. For a cipher to be considered secure, this bound must be sufficiently large.

2.3 Differential and Linear Branch Number of a Matrix

The differential branch number [Dae95, DR02] of a matrix $M : (\mathbb{F}_2^s)^m \to (\mathbb{F}_2^s)^m$ is the minimum summation of the Hamming weight of input difference (nonzero) and output difference, i.e.

$$B_d(M) = \min_{\Delta \neq 0}\{\mathrm{wt}(\Delta) + \mathrm{wt}(M\Delta)\},$$

where $\mathrm{wt}(X) = \#\{x_i \neq 0 | i \in \{0, \cdots, m-1\}\}$ for $X = (x_0, x_1, \cdots, x_{m-1})$ represents the Hamming weight of X. Similarly, the linear branch number [Dae95, DR02] of a matrix M is

$$B_l(M) = \min_{\Gamma \neq 0}\{\mathrm{wt}(\Gamma) + \mathrm{wt}(M^T\Gamma)\},$$

where M^T is the transpose of the M. If we say that the branch number of a matrix is n, we mean that both its differential and linear branch numbers are at least n, and that one of them is exactly equal to n. If the branch number of M achieves the maximum $m+1$, M is called a maximum distance separable (MDS) matrix. And M is called an almost MDS matrix if the branch number is m.

2.4 The Method of Classifying Shuffle Layers in [ABI+18]

In [ABI+18], Gianira Alfarano et al. introduced the notion of *M-equivalence* to classify shuffle layers at the cell level. Consider an AES-like cipher (see Equation 1), where the state is represented as an $r \times c$ matrix of cells. Let $\mathbb{S}_{rc}$ denote the set of all permutations over $\{0, 1, \ldots, rc-1\}$, and let $\mathcal{T}(M)$ denote the set of all cell permutations $\mathtt{Q} \in \mathbb{S}_{rc}$ that commute with the MixColumns operation $\mathtt{MC}$, where M is the diffusion matrix used in $\mathtt{MC}$.

Definition 1 (M-equivalence)**.** Let $(\mathbb{F}_2^s)^r \to (\mathbb{F}_2^s)^r$. We say that two shuffle layers $\mathtt{P}$ and $\mathtt{P}'$ are **M-equivalent** if there exists a permutation $\mathtt{Q} \in \mathcal{T}(M)$ such that

$$\mathtt{P}' = \mathtt{Q} \circ \mathtt{P} \circ \mathtt{Q}^{-1}.$$

We write $\mathtt{P} \sim_M \mathtt{P}'$ when $\mathtt{P}$ and $\mathtt{P}'$ are M-equivalent.

3 Differential Pattern Transition

For an input (resp. output) difference $\Delta \in (\mathbb{F}_2^s)^m$, we use an array in $\{0,1\}^m$ to indicate whether each s-bit cell in Δ is active. This array is referred to as the input (resp. output) differential pattern, where a value of 1 denotes an nonzero cell and a value of 0 corresponds to an all-zero cell. Given a diffusion matrix $M : (\mathbb{F}_2^s)^m \to (\mathbb{F}_2^s)^m$, there exists a natural correspondence between its input and output differential patterns. We introduce this correspondence as a new cryptographic notion for the diffusion matrix, called differential pattern transition, abbreviated as DPT (see Definition 2).

Table 1: The DPT of M_{midori}.

Input pattern	Output pattern
0000	0000
0001	1110
0010	1101
0011	0011 1111
0100	1011
0101	0101 1111
0110	0110 1111
0111	1000 0111 1011 1101 1110 1111
1000	0111
1001	1001 1111
1010	1010 1111
1011	0100 0111 1011 1101 1110 1111
1100	1100 1111
1101	0010 0111 1011 1101 1110 1111
1110	0001 0111 1011 1101 1110 1111
1111	0011 0101 0110 1001 1010 1100 0111 1011 1101 1110 1111

Definition 2 (Differential pattern transition)**.** The *differential pattern transition* of a diffusion matrix M is defined as the set of all possible transitions from input differential patterns to output differential patterns induced by M, denoted by DPT_M. For a given input pattern i, the set of all possible output patterns is denoted by $\mathsf{DPT}_M[i]$.

The DPT has previously been employed to model cell-wise differential propagation of `AES`-like cipher [MWGP12, SHW+14, Ava17], but, to the best of our knowledge, it has not been formally defined. In this work, we extend its application and show in the next sections that DPT allows us to distinguish the strength of diffusion matrices against differential attacks, even when they share the same branch number. Furthermore, it can aid in classifying shuffle layers and provides a practical tool for searching permutation layers that optimize differential resistance.

Example 1. Table 1 provides an example of the DPT corresponding to the diffusion matrix over $\mathbb{F}_2^4$ used in `Midori`, denoted by M_{midori}:

$$M_{\text{midori}} = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}. \tag{2}$$

For any input pattern i, the corresponding set $\mathsf{DPT}_{M_{\text{midori}}}[i]$ can be obtained directly from Table 1. For instance, $\mathsf{DPT}_{M_{\text{midori}}}[0011] = \{0011, 1111\}$.

Conversely, one may regard a table containing input and output patterns as a DPT, even if it does not correspond to any concrete diffusion matrix. For convenience of exposition, we assume that each DPT is associated with a *dummy diffusion matrix*. This matrix is

neither solved for nor utilized in our analysis; it merely serves as a notational device to represent the diffusion matrix corresponding to a given DPT.

It is worth noting that, in an analogous manner, one can define a related cryptographic notion for linear attacks, referred to as linear pattern transition (see Definition 3). In fact, the linear pattern transition of a diffusion matrix M is equivalent to the differential pattern transition of $(M^T)^{-1}$. Hence, the properties established for differential pattern transition can be analogously extended to linear pattern transition. Therefore, throughout this paper, we restrict our attention to differential pattern transition.

Definition 3 (Linear pattern transition)**.** The *linear pattern transition* of a diffusion matrix M is defined as the set of all possible transitions from input linear patterns to output linear patterns induced by M, denoted by LPT_M.

3.1 Partial Order Relation

Intuitively, a denser DPT indicates a larger number of possible differential characteristics, which in turn results in a smaller minimal number of active S-boxes. To enable meaningful comparison of the minimal active S-box counts across different matrices, we introduce a partial order relation on DPTs, which provides a foundation for comparing the relative strength of diffusion matrices in resisting differential attacks. In Subsection 4.1, we further discuss how this partial order relates to differential resistance. Moreover, since the partial order relation enables a formal comparison among diffusion matrices, and the branch number serves as another classical metric for diffusion strength, we also investigate the relationship between this partial order and the branch number.

Definition 4 (Partial order relation of DPTs)**.** Let DPT_1 and DPT_2 be two DPTs. We say $\mathsf{DPT}_1 \leq \mathsf{DPT}_2$ if and only if

$$\forall \text{ input pattern } i, \mathsf{DPT}_1[i] \subseteq \mathsf{DPT}_2[i].$$

It is straightforward to verify that this relation satisfies the following properties:

- Reflexivity: $\mathsf{DPT}_1 \leq \mathsf{DPT}_1$.
- Antisymmetry: if $\mathsf{DPT}_1 \leq \mathsf{DPT}_2$ and $\mathsf{DPT}_2 \leq \mathsf{DPT}_1$ then $\mathsf{DPT}_1 = \mathsf{DPT}_2$.
- Transitivity: if $\mathsf{DPT}_1 \leq \mathsf{DPT}_2$ and $\mathsf{DPT}_2 \leq \mathsf{DPT}_3$ then $\mathsf{DPT}_1 \leq \mathsf{DPT}_3$.

Therefore, the partial order [Tro92] of DPTs is well defined.

It should be emphasized that not every pair of diffusion matrices admits a partial order with respect to their DPTs.

We now investigate how the partial order relation between DPTs interacts with the branch number.

Lemma 1 (Per-input monotonicity)**.** *For any diffusion matrices M_1, M_2 and any input pattern i, if $\mathsf{DPT}_{M_1}[i] \subseteq \mathsf{DPT}_{M_2}[i]$, then*

$$\min_{o \in \mathsf{DPT}_{M_1}[i]} \mathrm{wt}(o) \;\geq\; \min_{o \in \mathsf{DPT}_{M_2}[i]} \mathrm{wt}(o).$$

Proposition 1 (Monotonicity of differential branch number under the DPT order)**.** *Let M_1, M_2 be diffusion matrices. If $\mathsf{DPT}_{M_1} \leq \mathsf{DPT}_{M_2}$, then*

$$B_d(M_1) \;\geq\; B_d(M_2).$$

Proof. By Lemma 1, for every input pattern i, $\min_{o \in \mathsf{DPT}_{M_1}[i]} \mathrm{wt}(o) \geq \min_{o \in \mathsf{DPT}_{M_2}[i]} \mathrm{wt}(o)$. Adding $\mathrm{wt}(i)$ to both sides and minimizing over $i \neq 0$ yields $B_d(M_1) \geq B_d(M_2)$. □

Corollary 1 (Direction of comparability)**.** *If $B(M_1) < B(M_2)$, then $\mathsf{DPT}_{M_1} \leq \mathsf{DPT}_{M_2}$ is impossible. If a partial-order relation exists, it must be $\mathsf{DPT}_{M_2} \leq \mathsf{DPT}_{M_1}$. Symmetrically, if $B(M_1) > B(M_2)$, any possible relation must be $\mathsf{DPT}_{M_1} \leq \mathsf{DPT}_{M_2}$.*

Different branch numbers do not guarantee the comparability of the corresponding DPTs; once they are comparable, however, their relative order is determined. Likewise, identical branch numbers do not ensure comparability, and even when the DPTs are comparable, the direction of this order cannot be inferred.

3.2 Matrix-DPT-equivalence of Permutation Pairs

For an AES-like cipher (see Equation (1)), the shuffle layer P can be interchanged with either the S-box layer S or the key addition AK. Specifically, adding a key k before applying P is equivalent to adding $\mathtt{P}(k)$ after the permutation. As a result of such rearrangement, the MixColumns operation MC becomes surrounded by shuffle layers on both sides.

In [ABI+18], the authors investigated the class of shuffle layers that commute with MC, denoted as $\mathcal{T}(M)$, and introduced the notion of M-equivalence based on cipher equivalence (see Definition 1). Building upon this idea, we jointly consider MC and the permutations applied on both sides, but from the perspective of differential behavior equivalence. Consequently, it is natural to analyze the combined effect of the diffusion matrix M and the surrounding permutations. To this end, we define an equivalence relation on pairs of permutations in terms of the DPT induced by M. This relation will later be employed in Subsection 4.2 for the classification of shuffle layers.

Let $\mathbb{S}_m$ denote the set of all permutations over $\{0, 1, \ldots, m-1\}$, and let $M : (\mathbb{F}_2^s)^m \to (\mathbb{F}_2^s)^m$ be a fixed diffusion matrix. For $\mathtt{p}_1, \mathtt{p}_2, \mathtt{p}_1', \mathtt{p}_2' \in \mathbb{S}_m$, we make the following definition.

Definition 5 (Matrix-DPT-equivalence of permutation pairs)**.** We say that the pairs $(\mathtt{p}_1, \mathtt{p}_2)$ and $(\mathtt{p}_1', \mathtt{p}_2')$ are *matrix-DPT-equivalent* with respect to M if the DPTs of $\mathtt{p}_2 \circ M \circ \mathtt{p}_1$ and $\mathtt{p}_2' \circ M \circ \mathtt{p}_1'$ coincide. Formally,

$$(\mathtt{p}_1, \mathtt{p}_2) \overset{\mathsf{DPT}}{\sim}_M (\mathtt{p}_1', \mathtt{p}_2') \quad \Longleftrightarrow \quad \mathsf{DPT}_{\mathtt{p}_2 \circ M \circ \mathtt{p}_1} = \mathsf{DPT}_{\mathtt{p}_2' \circ M \circ \mathtt{p}_1'}.$$

It is straightforward to verify matrix-DPT-equivalence of permutation pairs is an equivalence relation, i.e., it is reflexive, symmetric, and transitive.

In our study, we identify certain diffusion matrices exhibiting special Matrix-DPT-equivalence properties and introduce the following definitions.

Definition 6 (Permutation-invariant (PI) matrix)**.** A diffusion matrix M is called a *permutation-invariant matrix* (PI matrix) if any pairs $(\mathtt{p}_1, \mathtt{p}_2)$ and $(\mathtt{p}_1', \mathtt{p}_2')$ are matrix-DPT-equivalent with respect to M.

Many matrices fall into this category, such as the diffusion matrix of AES. For a PI matrix, the number of matrix-DPT-equivalence classes is 1, and the representative can be chosen as $(\mathtt{id}, \mathtt{id})$.

Definition 7 (Permutation-commutative (PC) matrix)**.** A diffusion matrix M is called a *permutation-commutative matrix* (PC matrix) if any pairs $(\mathtt{p}_1, \mathtt{p}_2)$ and $(\mathtt{p}_1', \mathtt{p}_2')$ satisfying $\mathtt{p}_2 \circ \mathtt{p}_1 = \mathtt{p}_2' \circ \mathtt{p}_1'$ are matrix-DPT-equivalent with respect to M.

An example of a PC matrix is the diffusion matrix of MIDORI. The set of representatives can be chosen as $\{(\mathtt{p}_1, \mathtt{id}) \mid \mathtt{p}_1 \in \mathbb{S}_m\}$, and the number of equivalence classes is $m!$.

Definition 8 (Rotation-commutative (RC) matrix)**.** A diffusion matrix M is called a *rotation-commutative matrix* (RC matrix) if for every circular shift operation sr, the permutation pairs $(\mathtt{p}_1, \mathtt{p}_2)$ and $(\mathtt{sr} \circ \mathtt{p}_1,\ \mathtt{p}_2 \circ \mathtt{sr}^{-1})$ are matrix-DPT-equivalent with respect to M. Here, sr denotes a rotation on $\mathbb{Z}_m$ defined by

$$i \overset{\mathtt{sr}}{\mapsto} (i + j) \bmod m, \qquad \text{for a fixed } j \in \mathbb{Z}_m.$$

Circulant matrices are a natural example of RC matrices. Since there are m possible circular shift operations, the number of equivalence classes is $m!(m-1)!$.

By definition, every PI matrix is a PC matrix, and every PC matrix is an RC matrix. Moreover, each Matrix-DPT-equivalence class for an RC matrix is contained in one for a PC matrix, which in turn is contained in one for a PI matrix.

4 Differential Pattern Transition in AES-like Ciphers

In this section, we show how DPT can distinguish the strength of diffusion matrices against differential attacks, while also aiding in the classification of shuffle layers and the search for permutation layers with optimal differential resistance.

4.1 Connecting Diffusion Matrices to Differential Characteristics

We will show that DPTs naturally establish a connection between a diffusion matrix and the differential characteristics of the AES-like SPN cipher. Subsequently, we demonstrate that the partial order defined on DPTs provides a way to compare the lower bounds on the number of active S-boxes in differential characteristics.

Recall that the structure of an AES-like SPN cipher can be described as

$$\mathtt{S} \to \mathtt{P} \to \mathtt{MC} \to \mathtt{AK} \to \mathtt{S} \to \mathtt{P} \to \mathtt{MC} \to \mathtt{AK} \to \cdots \tag{3}$$

Since the S-box layer S commutes with the cell shuffle layer P, the S operation and the MC operation can be viewed together as a *composite S-box* applied to each column, denoted by MS. Thus, the cipher can be equivalently rewritten as

$$\mathtt{P} \to \mathtt{MS} \to \mathtt{AK} \to \mathtt{P} \to \mathtt{MS} \to \mathtt{AK} \to \cdots \tag{4}$$

where $\mathtt{MS} = \mathtt{MC} \circ \mathtt{S}$.

In this paper, we assume that the S-box in the S layer can be any nonlinear bijective vectorial Boolean function, and that the S-boxes in different cells are not required to be identical. Under this assumption, the DPT of the diffusion matrix in the MC operation are equivalent to cell-wise input–output difference transitions for the composite S-box in the MS operation. Consequently, the DPTs of the diffusion matrix alone are sufficient to describe the cell-level differential characteristics of the cipher.

Theorem 1. *Let E_1 and E_2 be two ciphers with AES-like SPN structure as in Equation (3), where the P layers are identical. Let the DPTs of the diffusion matrices in the MC operation be DPT_1 and DPT_2, respectively. If $\mathsf{DPT}_1 \le \mathsf{DPT}_2$, then the lower bound on the number of active S-boxes for n-round differential characteristics of E_1 is greater than or equal to that of E_2.*

Proof. Consider the n-round trail at the cell level for E_1 in the form

$$A_1 \xrightarrow{\mathtt{P}} B_1 \xrightarrow{\mathtt{MS}} A_2 \xrightarrow{\mathtt{AK}} A_2 \xrightarrow{\mathtt{P}} B_2 \xrightarrow{\mathtt{MS}} A_3 \xrightarrow{\mathtt{AK}} \cdots \xrightarrow{\mathtt{AK}} A_{n+1},$$

where the initial state A_1 contains at least one nonzero cell. For each column, the propagation satisfies

$$A_{i+1,j} \in \mathsf{DPT}_1[B_{i,j}],$$

where $A_{i+1,j}$ and $B_{i,j}$ denote the j-th column of A_{i+1} and B_i, respectively.

Since $\mathsf{DPT}_1 \le \mathsf{DPT}_2$, every such transition is also valid for E_2. Hence, every differential trail possible for E_1 is also a valid trail for E_2, although E_2 may admit additional trails not possible in E_1. Therefore, the lower bound on the number of active S-boxes in n-round differential characteristics of E_1 is greater than or equal to that of E_2. □

4.2 Classifying the Shuffle Layer

In the design of AES-like SPN ciphers, resistance against differential cryptanalysis is a crucial security criterion. Since the shuffle layer P plays a key role in diffusion, its proper design directly affects the lower bound on the number of active S-boxes. In this subsection, we study the classification of shuffle layers P in the AES-like SPN structure of Equation 3. Specifically, for a fixed diffusion matrix M, we show that by employing DPTs, the set of all possible shuffle layers can be classified into equivalence classes such that every shuffle layer within the same class yields the same lower bound on the number of active S-boxes in the cipher's differential characteristics.

4.2.1 Decomposition of Shuffle Layers

We begin by decomposing the shuffle layer, and subsequently employ this decomposition as the basis for its classification.

Assume that the cipher state in Equation 3 is represented as an $r \times c$ matrix over $\mathbb{F}_2^s$. The MC operation can then be viewed as a linear diffusion mapping

$$\mathtt{MC} : (\mathbb{F}_2^s)^{rc} \longrightarrow (\mathbb{F}_2^s)^{rc}.$$

Building on the concept of the *structure matrix* introduced in [ABI+18], we employ it to achieve a systematic decomposition of the shuffle layer in our study. The structure matrix of a shuffle layer $\mathtt{P} \in \mathbb{S}_{rc}$ is defined as the $c \times c$ matrix $A_{\mathtt{P}}$, where the entry $A_{\mathtt{P}}[i,j]$ denotes the number of cells from column i that are permuted into column j under P. Formally,

$$A_{\mathtt{P}}[i,j] \;=\; \Big|\{\, k \in \mathbb{Z}_{rc} \;\mid\; k = ri + l,\; l \in \mathbb{Z}_r,\; ri + l \xrightarrow{\mathtt{P}} rj + q,\; q \in \mathbb{Z}_r \}\Big|.$$

We also define the set

$$\mathbb{PC} := \{\, \mathtt{P} \in \mathbb{S}_{rc} \;\mid\; \exists \mathtt{p}_0, \ldots, \mathtt{p}_{c-1} \in \mathbb{S}_r \text{ such that } \forall i \in \mathbb{Z}_r,\; j \in \mathbb{Z}_c :\; rj + i \xrightarrow{\mathtt{P}} rj + \mathtt{p}_j(i) \,\}.$$

That is, $\mathbb{PC}$ consists of permutations that only permute within each column.

It is clear that the structure matrix induces a partition of the entire set of shuffle layers, i.e., shuffle layers in the same *partition class* share the same structure matrix.

Now let $\mathtt{P}_1$ and $\mathtt{P}_2$ be two shuffle layers belonging to the same partition class. Then for any $i, j \in \mathbb{Z}_c$, we have

$$A_{\mathtt{P}_1}[i,j] \;=\; A_{\mathtt{P}_2}[i,j].$$

In particular, there exist distinct indices $k_0, \ldots, k_{A_{\mathtt{P}_1}[i,j]-1}$ such that for every $0 \le d < A_{\mathtt{P}_1}[i,j]$,

$$k_d = ri + l_d \xrightarrow{\mathtt{P}_1} rj + q_d, \qquad l_d, q_d \in \mathbb{Z}_r,$$

and likewise, there exist distinct indices k'_ds such that

$$k'_d = ri + l'_d \xrightarrow{\mathtt{P}_2} rj + q'_d, \qquad l'_d, q'_d \in \mathbb{Z}_r.$$

For all $i \in \mathbb{Z}_c$, the correspondence between $\{l_d\}$ and $\{l'_d\}$ defines a unique global permutation $\mathtt{PC}_2 \in \mathbb{PC}$ such that $ri + l_d \mapsto ri + l'_d$ for every d. Similarly, for all $j \in \mathbb{Z}_c$, the correspondence between $\{q'_d\}$ and $\{q_d\}$ defines a unique global permutation $\mathtt{PC}_1 \in \mathbb{PC}$ such that $rj + q'_d \mapsto rj + q_d$ for every d. Hence we obtain

$$\mathtt{P}_1 = \mathtt{PC}_1 \circ \mathtt{P}_2 \circ \mathtt{PC}_2. \tag{5}$$

We should note that if there exists some i, j such that $A_{\mathtt{P}_1}[i,j] > 1$, then the sequences $\{k_0, \ldots, k_{A_{\mathtt{P}_1}[i,j]-1}\}$ and $\{k'_0, \ldots, k'_{A_{\mathtt{P}_1}[i,j]-1}\}$ may be rearranged. This could lead to different choices of permutations $\mathtt{PC}'_1, \mathtt{PC}'_2 \in \mathbb{PC}$ such that still

$$\mathtt{P}_1 = \mathtt{PC}'_1 \circ \mathtt{P}_2 \circ \mathtt{PC}'_2. \tag{6}$$

By combining Equations (5) and (6), we obtain

$$\mathtt{P}_2 = (\mathtt{PC}_1)^{-1} \circ \mathtt{PC}_1' \circ \mathtt{P}_2 \circ \mathtt{PC}_2' \circ (\mathtt{PC}_2)^{-1}.$$

Therefore, once we know one representation of $\mathtt{P}_1$ with respect to $\mathtt{P}_2$ as in Equation (5), all other valid representations can be derived from the possible representations of $\mathtt{P}_2$ with respect to itself.

The self-representations of $\mathtt{P}_2$ correspond to all possible bijections of the sequences $\{l_d'\}$ to themselves for each $i, j \in \mathbb{Z}_c$, while simultaneously mapping the corresponding sequences $\{q_d'\}$ to themselves. Hence, the total number of possible representations of $\mathtt{P}_2$ is given by

$$\prod_{\substack{i,j \\ A_{\mathtt{P}_2}[i,j]\neq 0}} \big(A_{\mathtt{P}_2}[i,j]!\big),$$

where $A_{\mathtt{P}_2}[i,j]!$ denotes the factorial of $A_{\mathtt{P}_2}[i,j]$.

Example 2. Suppose

$$\begin{bmatrix} 0 & 2 \\ 1 & 3 \end{bmatrix} \xrightarrow{\mathtt{P}} \begin{bmatrix} 2 & 1 \\ 3 & 0 \end{bmatrix},$$

then the structure matrix is

$$A_{\mathtt{P}} = \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix},$$

so the total number of possible representations is $2! \times 2! = 4$. These decompositions are summarized in Table 2.

Table 2: All 4 possible representations of $\mathtt{P}$.

Case	Representation
1	$\begin{bmatrix} 0 & 2 \\ 1 & 3 \end{bmatrix} \xrightarrow{\mathtt{P}} \begin{bmatrix} 2 & 1 \\ 3 & 0 \end{bmatrix}$
2	$\begin{bmatrix} 0 & 2 \\ 1 & 3 \end{bmatrix} \xrightarrow{\mathtt{PC}_1} \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix} \xrightarrow{\mathtt{P}} \begin{bmatrix} 2 & 0 \\ 3 & 1 \end{bmatrix} \xrightarrow{\mathtt{PC}_2} \begin{bmatrix} 2 & 1 \\ 3 & 0 \end{bmatrix}$
3	$\begin{bmatrix} 0 & 2 \\ 1 & 3 \end{bmatrix} \xrightarrow{\mathtt{PC}_3} \begin{bmatrix} 1 & 3 \\ 0 & 2 \end{bmatrix} \xrightarrow{\mathtt{P}} \begin{bmatrix} 3 & 0 \\ 2 & 1 \end{bmatrix} \xrightarrow{\mathtt{PC}_4} \begin{bmatrix} 2 & 1 \\ 3 & 0 \end{bmatrix}$
4	$\begin{bmatrix} 0 & 2 \\ 1 & 3 \end{bmatrix} \xrightarrow{\mathtt{PC}_5} \begin{bmatrix} 0 & 3 \\ 1 & 2 \end{bmatrix} \xrightarrow{\mathtt{P}} \begin{bmatrix} 3 & 1 \\ 2 & 0 \end{bmatrix} \xrightarrow{\mathtt{PC}_6} \begin{bmatrix} 2 & 1 \\ 3 & 0 \end{bmatrix}$

Proposition 2. *Any shuffle layer* $\mathtt{P}$ *can be expressed in the form*

$$\mathtt{P} \;=\; \mathtt{PC}_1 \circ \mathtt{P}_r \circ \mathtt{PC}_2,$$

where $\mathtt{P}_r$ *is the chosen representative of the partition class containing* $\mathtt{P}$*, and* $\mathtt{PC}_1, \mathtt{PC}_2 \in \mathbb{PC}$*. Moreover, the total number of such representations, denoted by* $N_{\mathtt{P}_r}$*, given by*

$$N_{\mathtt{P}_r} := \prod_{\substack{i,j\in\mathbb{Z}_c \\ A_{\mathtt{P}_r}[i,j]\neq 0}} \big(A_{\mathtt{P}_r}[i,j]!\big),$$

where $A_{\mathtt{P}_r}$ *denotes the structure matrix of* $\mathtt{P}_r$*.*

4.2.2 The Classification of Shuffle Layers

In this subsection, we formally introduce the notion of MC-DPT-equivalence of shuffle layers. We then employ this equivalence relation, together with certain properties of M-equivalence (see Definition 1), to classify the shuffle layers.

Definition 9 (MC-DPT-equivalence of shuffle layers)**.** Let $\mathtt{P}$ and $\mathtt{P}'$ be two shuffle layers sharing the same structure matrix. We say that $\mathtt{P}$ and $\mathtt{P}'$ are *MC-DPT-equivalence* with respect to $\mathtt{MC}$ if there exist $\mathtt{PC}_1, \mathtt{PC}_2, \mathtt{PC}'_1, \mathtt{PC}'_2 \in \mathbb{PC}$ such that

$$\mathtt{P} = \mathtt{PC}_1 \circ \mathtt{P}_r \circ \mathtt{PC}_2, \quad \text{and} \quad \mathtt{P}' = \mathtt{PC}'_1 \circ \mathtt{P}_r \circ \mathtt{PC}'_2, \tag{7}$$

and moreover

$$\mathsf{DPT}_{\mathtt{PC}_2 \circ \mathtt{MC} \circ \mathtt{PC}_1} = \mathsf{DPT}_{\mathtt{PC}'_2 \circ \mathtt{MC} \circ \mathtt{PC}'_1}. \tag{8}$$

In this case we write $\mathtt{P} \overset{\mathsf{DPT}}{\sim}_{\mathtt{MC}} \mathtt{P}'$.

Lemma 2. *MC-DPT-equivalence is independent of the choice of the partition class representative.*

Proof. Assume

$$\mathtt{P} \overset{\mathsf{DPT}}{\sim}_{\mathtt{MC}} \mathtt{P}',$$

and let $\mathtt{P}$ and $\mathtt{P}'$ satisfy the decompositions of Equation 7 and the DPT equality of Equation 8. Now let $\tilde{\mathtt{P}}_r$ be another element of the same partition class as $\mathtt{P}_r$. By definition of the partition class, there exist permutation cells $\tilde{\mathtt{PC}}_1, \tilde{\mathtt{PC}}_2 \in \mathbb{PC}$ such that

$$\mathtt{P}_r = \tilde{\mathtt{PC}}_1 \circ \tilde{\mathtt{P}}_r \circ \tilde{\mathtt{PC}}_2.$$

Substituting this expression into the decompositions of $\mathtt{P}$ and $\mathtt{P}'$ yields new decompositions

$$\mathtt{P} = (\mathtt{PC}_1 \circ \tilde{\mathtt{PC}}_1) \circ \tilde{\mathtt{P}}_r \circ (\tilde{\mathtt{PC}}_2 \circ \mathtt{PC}_2), \qquad \mathtt{P}' = (\mathtt{PC}'_1 \circ \tilde{\mathtt{PC}}_1) \circ \tilde{\mathtt{P}}_r \circ (\tilde{\mathtt{PC}}_2 \circ \mathtt{PC}'_2).$$

By inserting $\tilde{\mathtt{PC}}_1$ and $\tilde{\mathtt{PC}}_2$ into both sides of (8), we obtain

$$\mathsf{DPT}_{\tilde{\mathtt{PC}}_2 \circ \mathtt{PC}_2 \circ \mathtt{MC} \circ \mathtt{PC}_1 \circ \tilde{\mathtt{PC}}_1} = \mathsf{DPT}_{\tilde{\mathtt{PC}}_2 \circ \mathtt{PC}'_2 \circ \mathtt{MC} \circ \mathtt{PC}'_1 \circ \tilde{\mathtt{PC}}_1}.$$

This equality is exactly the condition required by Definition 9 for the decompositions using the alternative representative $\tilde{\mathtt{P}}_r$.

Therefore, the definition of MC-DPT-equivalence is independent of the choice of the partition class representative. □

Theorem 2. *The MC-DPT-equivalence of shuffle layers is an equivalence relation.*

Proof. Reflexivity and symmetry follow directly from Definition 9. We prove transitivity.

Assume

$$\mathtt{P} \overset{\mathsf{DPT}}{\sim}_{\mathtt{MC}} \mathtt{P}' \qquad \text{and} \qquad \mathtt{P} \overset{\mathsf{DPT}}{\sim}_{\mathtt{MC}} \mathtt{P}''.$$

By Definition 9, there exist permutation cells $\mathtt{PC}_1, \mathtt{PC}_2, \mathtt{PC}'_1, \mathtt{PC}'_2$ such that

$$\mathtt{P} = \mathtt{PC}_1 \circ \mathtt{P}_r \circ \mathtt{PC}_2, \qquad \mathtt{P}' = \mathtt{PC}'_1 \circ \mathtt{P}_r \circ \mathtt{PC}'_2,$$

and

$$\mathsf{DPT}_{\mathtt{PC}_2 \circ \mathtt{MC} \circ \mathtt{PC}_1} = \mathsf{DPT}_{\mathtt{PC}'_2 \circ \mathtt{MC} \circ \mathtt{PC}'_1}. \tag{9}$$

For the second equivalence, there exist permutation cells $\tilde{\mathtt{PC}}_1, \tilde{\mathtt{PC}}_2$ and $\mathtt{PC}''_1, \mathtt{PC}''_2$ such that

$$\mathtt{P} = \tilde{\mathtt{PC}}_1 \circ \mathtt{P}_r \circ \tilde{\mathtt{PC}}_2, \qquad \mathtt{P}'' = \mathtt{PC}''_1 \circ \mathtt{P}_r \circ \mathtt{PC}''_2,$$

and

$$\mathsf{DPT}_{\tilde{\mathtt{PC}}_2 \circ \mathtt{MC} \circ \tilde{\mathtt{PC}}_1} = \mathsf{DPT}_{\mathtt{PC}''_2 \circ \mathtt{MC} \circ \mathtt{PC}''_1}. \tag{10}$$

Define

$$\widehat{\mathtt{PC}}_1 = (\mathtt{PC}_1)^{-1} \circ \tilde{\mathtt{PC}}_1, \qquad \widehat{\mathtt{PC}}_2 = \tilde{\mathtt{PC}}_2 \circ (\mathtt{PC}_2)^{-1},$$

so that

$$\tilde{\mathtt{PC}}_1 = \mathtt{PC}_1 \circ \widehat{\mathtt{PC}}_1, \qquad \tilde{\mathtt{PC}}_2 = \widehat{\mathtt{PC}}_2 \circ \mathtt{PC}_2.$$

Substituting these expressions into (10) and using (9), we obtain

$$\mathsf{DPT}_{\mathtt{PC}''_2 \circ \mathtt{MC} \circ \mathtt{PC}''_1} = \mathsf{DPT}_{\widehat{\mathtt{PC}}_2 \circ \mathtt{PC}'_2 \circ \mathtt{MC} \circ \mathtt{PC}'_1 \circ \widehat{\mathtt{PC}}_1}.$$

Since

$$\widehat{\mathtt{PC}}_1 \circ \mathtt{P}_r \circ \widehat{\mathtt{PC}}_2$$

is a representation of $\mathtt{P}_r$, the expression

$$\mathtt{PC}'_1 \circ \widehat{\mathtt{PC}}_1 \ \circ \ \mathtt{P}_r \ \circ \ \widehat{\mathtt{PC}}_2 \circ \mathtt{PC}'_2$$

is a valid decomposition of $\mathtt{P}'$. Thus, by Definition 9, the above equality shows that

$$\mathtt{P}' \overset{\mathsf{DPT}}{\sim}_{\mathtt{MC}} \mathtt{P}''.$$

Therefore, MC-DPT-equivalence is an equivalence relation. □

Theorem 3. *Consider two ciphers based on the AES-like SPN structure:*

(1) $\mathtt{S} \to \mathtt{P} \to \mathtt{MC} \to \mathtt{AK} \to \mathtt{S} \to \mathtt{P} \to \mathtt{MC} \to \mathtt{AK} \cdots$,

(2) $\mathtt{S} \to \mathtt{P}' \to \mathtt{MC} \to \mathtt{AK} \to \mathtt{S} \to \mathtt{P}' \to \mathtt{MC} \to \mathtt{AK} \cdots$,

where $\mathtt{P}$ *and* $\mathtt{P}'$ *are two shuffle layers. If* $\mathtt{P} \overset{\mathsf{DPT}}{\sim}_{\mathtt{MC}} \mathtt{P}'$*, then there exists a one-to-one correspondence between the cell-level differential characteristics of the two ciphers. In particular, the lower bound on the number of active S-boxes in their differential characteristics is identical.*

Proof. By assumption, $\mathtt{P} \overset{\mathsf{DPT}}{\sim}_{\mathtt{MC}} \mathtt{P}'$, i.e., $\mathtt{P}$ and $\mathtt{P}'$ share the same structure matrix. Let

$$\mathtt{P} = \mathtt{PC}_1 \circ \mathtt{P}_r \circ \mathtt{PC}_2 \quad \text{and} \quad \mathtt{P}' = \mathtt{PC}'_1 \circ \mathtt{P}_r \circ \mathtt{PC}'_2,$$

and we have

$$\mathsf{DPT}_{\mathtt{PC}_2 \circ \mathtt{MC} \circ \mathtt{PC}_1} \ = \ \mathsf{DPT}_{\mathtt{PC}'_2 \circ \mathtt{MC} \circ \mathtt{PC}'_1}.$$

Let an n-round cell-level differential trail of the first cipher be represented as a sequence of states

$$A_1 \xrightarrow{\mathtt{P}} B_1 \xrightarrow{\mathtt{S}} B_1 \xrightarrow{\mathtt{MC}} A_2 \xrightarrow{\mathtt{AK}} A_2 \xrightarrow{\mathtt{P}} B_2 \xrightarrow{\mathtt{S}} B_2 \xrightarrow{\mathtt{MC}} \cdots \xrightarrow{\mathtt{AK}} A_{n+1},$$

where each arrow denotes a possible cell-wise transition (with nonzero probability) in the corresponding operation.

Since $\mathtt{P} = \mathtt{PC}_1 \circ \mathtt{P}_r \circ \mathtt{PC}_2$, this trail is equivalent to

$$A_1 \overset{\mathtt{PC}_2}{\to} C_1 \overset{\mathtt{P}_r}{\to} \underbrace{D_i \overset{\mathtt{S}}{\to} D_i \overset{\mathtt{PC}_1}{\to} B_i \overset{\mathtt{MC}}{\to} A_{i+1} \overset{\mathtt{PC}_2}{\to} C_{i+1} \overset{\mathtt{P}_r}{\to} D_{i+1} \overset{\mathtt{AK}}{\to} D_{i+1}}_{\text{for } i=1,\ldots,n} \overset{\mathtt{P}_r^{-1}}{\to} C_{n+1} \overset{\mathtt{PC}_2^{-1}}{\to} A_{n+1},$$

where $C_i = \mathtt{PC}_2(A_i)$ and $D_i = \mathtt{P}_r(C_i)$.

By the definition of the differential pattern transition, for every round index i we have

$$C_{i+1} \in \mathsf{DPT}_{\mathtt{PC}_2 \circ \mathtt{MC} \circ \mathtt{PC}_1}[D_i].$$

Since

$$\mathsf{DPT}_{\mathtt{PC}_2 \circ \mathtt{MC} \circ \mathtt{PC}_1} = \mathsf{DPT}_{\mathtt{PC}'_2 \circ \mathtt{MC} \circ \mathtt{PC}'_1},$$

the same membership conditions hold with $\mathtt{PC}'_1$ and $\mathtt{PC}'_2$ in place of $\mathtt{PC}_1$ and $\mathtt{PC}_2$, i.e.,

$$C_{i+1} \in \mathsf{DPT}_{\mathtt{PC}'_2 \circ \mathtt{MC} \circ \mathtt{PC}'_1}[D_i], \quad \forall i.$$

Therefore, we obtain a corresponding trail for the second cipher:

$$A'_1 \overset{\mathtt{PC}'_2}{\to} C_1 \overset{\mathtt{P}_\tau}{\to} \underbrace{D_i \overset{\mathtt{S}}{\to} D_i \overset{\mathtt{PC}'_1}{\to} B'_i \overset{\mathtt{MC}}{\to} A'_{i+1} \overset{\mathtt{PC}'_2}{\to} C_{i+1} \overset{\mathtt{P}_\tau}{\to} D_{i+1} \overset{\mathtt{AK}}{\to} D_{i+1}}_{\text{for } i=1,\ldots,n} \overset{(\mathtt{P}_r)^{-1}}{\to} C_{n+1} \overset{(\mathtt{P}'_2)^{-1}}{\to} A'_{n+1},$$

where $A'_i = (\mathtt{PC}'_2)^{-1}(C_i)$ and $B'_i = \mathtt{PC}'_1(D_i)$.

Thus, every valid trail of the first cipher corresponds to a valid trail of the second cipher. By symmetry (interchanging the roles of $\mathtt{P}$ and $\mathtt{P}'$), the converse also holds. Hence, the differential trails of the two ciphers are in one-to-one correspondence.

In particular, since the number of active S-boxes is given by $\sum_{i=1}^{n} \mathrm{wt}(D_i)$, the minimal number of active S-boxes over any n-round differential characteristic is identical for both ciphers. This completes the proof. □

Let $\mathbb{P}_{\rightleftharpoons}$ denote the set of all cell permutations that permute entire columns of the state:

$$\mathbb{P}_{\rightleftharpoons} := \{\mathtt{P} \in \mathbb{S}_{rc} \mid \exists \mathtt{p} \in \mathbb{S}_c : \forall i \in \mathbb{Z}_r,\, j \in \mathbb{Z}_c : rj + i \overset{\mathtt{P}}{\longmapsto} r\mathtt{p}(j) + i\},$$

where $\{rj + i\}_{i \in \mathbb{Z}_r}$ denotes the set of indices corresponding to the j-th column.

Since $\mathtt{MC}$ operates column-wise, it commutes with all permutations in $\mathbb{P}_{\rightleftharpoons}$. Thus, for any diffusion matrix M, we have

$$\mathbb{P}_{\rightleftharpoons} \subseteq \mathcal{T}(M).$$

Corollary 2. *For any diffusion matrix M, any $\mathtt{Q} \in \mathbb{P}_{\rightleftharpoons}$, and any $\mathtt{P} \in \mathbb{S}_{rc}$, let $\mathtt{P}' = \mathtt{Q} \circ \mathtt{P} \circ \mathtt{Q}^{-1}$. Then*

$$\mathtt{P} \sim_M \mathtt{P}'.$$

Since the construction of $\mathtt{P}'$ does not depend on M, we simply write $\mathtt{P} \rightleftharpoons \mathtt{P}'$.

Equivalence under $\sim_M$ ensures that the corresponding ciphers share identical differential properties. Since $\mathtt{MC}$-DPT-equivalence does not capture the commutativity of $\mathtt{MC}$ with all permutations in $\mathbb{P}_{\rightleftharpoons}$. We therefore refine the classification by incorporating the relation $\rightleftharpoons$ in addition to $\mathtt{MC}$-DPT-equivalence.

Concretely, for any $\mathtt{Q} \in \mathbb{P}_{\rightleftharpoons}$, there exists $\mathtt{q} \in \mathbb{S}_c$ such that

$$\forall i \in \mathbb{Z}_r,\, j \in \mathbb{Z}_c : \quad rj + i \overset{\mathtt{Q}}{\longmapsto} r\mathtt{q}(j) + i. \tag{11}$$

By Proposition 1 in [ABI+18], if $\mathtt{P}' = \mathtt{Q} \circ \mathtt{P} \circ \mathtt{Q}^{-1}$, then the corresponding structure matrices satisfy

$$A_{\mathtt{P}'}[i, j] := A_{\mathtt{P}}[\mathtt{q}^{-1}(i), \mathtt{q}^{-1}(j)].$$

We denote this as $A_{\mathtt{P}} \sim A_{\mathtt{P}'}$. Moreover, the elements of the partition class induced by $A_{\mathtt{P}}$ and those induced by $A_{\mathtt{P}'}$ are in one-to-one correspondence under the relation $\rightleftharpoons$. Hence, it suffices to partition all valid structure matrices into equivalence classes under $A_{\mathtt{P}} \sim A_{\mathtt{P}'}$ and then select a single representative structure matrix from each class for further representative shuffle layers filter.

Note that in some cases $A_{\mathtt{P}} = A_{\mathtt{P}'}$, meaning that redundancy may remain even after applying MC-DPT-equivalence. For such structure matrices, an additional refinement using the relation $\rightleftharpoons$ is required to obtain a minimal set of representatives.

Let $\mathtt{PC} \in \mathbb{PC}$ and let $\mathtt{Q}$ satisfy Eq. (11). Define

$$\mathtt{PC}^{\mathtt{q}} := \mathtt{Q} \circ \mathtt{PC} \circ \mathtt{Q}^{-1}.$$

It is easy to verify that $\mathtt{PC}^{\mathtt{q}} \in \mathbb{PC}$, and that the restriction of $\mathtt{PC}^{\mathtt{q}}$ to column $\mathtt{q}(j)$ is equivalent to the restriction of $\mathtt{PC}$ to column j.

Therefore, for a decomposition $\mathtt{P} = \mathtt{PC}_1 \circ \mathtt{P}_r \circ \mathtt{PC}_2$, we have

$$\mathtt{Q} \circ \mathtt{P} \circ \mathtt{Q}^{-1} = \mathtt{PC}_1^{\mathtt{q}} \ \circ \ (\mathtt{Q} \circ \mathtt{P}_r \circ \mathtt{Q}^{-1}) \ \circ \ \mathtt{PC}_2^{\mathtt{q}}.$$

Thus, it suffices to focus on the decomposition of $\mathtt{Q} \circ \mathtt{P}_r \circ \mathtt{Q}^{-1}$ in the filtering process under the relation $\rightleftharpoons$.

4.2.3 Construction of the Full Set of Representatives

We first classify all valid structure matrices under the relation $\sim$ and select one representative structure matrix from each equivalence class. Next, we describe how to further select representative shuffle layers within the partition class corresponding to each representative structure matrix.

By Theorem 3, we know that two shuffle layers $\mathtt{P}$ and $\mathtt{P}'$ that are MC-DPT-equivalence induce the same lower bound on the number of active S-boxes. Hence, when searching for good shuffle layers, it suffices to consider only one representative from each MC-DPT-equivalence class.

In practice, however, directly classifying all shuffle layers via the relation $\overset{\mathsf{DPT}}{\sim}_{\mathtt{MC}}$ (i.e. by comparing $\mathsf{DPT}_{\mathtt{PC}_2 \circ \mathtt{MC} \circ \mathtt{PC}_1}$ for every $\mathtt{P} = \mathtt{PC}_1 \circ \mathtt{P}_r \circ \mathtt{PC}_2$) is computationally expensive when the cipher state is large, because the number of permutations grows factorially.

Fortunately, since $\mathtt{PC}_2$, $\mathtt{MC}$, and $\mathtt{PC}_1$ are all column-wise operations, we can leverage the matrix-DPT-equivalence of permutation pairs introduced in Definition 5, which is typically easier to handle in practice, to efficiently construct the representatives of MC-DPT-equivalence classes.

Specifically, assume that

$$\mathsf{DPT}_{\mathtt{PC}_2 \circ \mathtt{MC} \circ \mathtt{PC}_1} \ = \ \mathsf{DPT}_{\mathtt{PC}'_2 \circ \mathtt{MC} \circ \mathtt{PC}'_1}. \tag{12}$$

Let $\mathtt{p}_{i,j}, \mathtt{p}'_{i,j} \in \mathbb{S}_r$ denote the restrictions of $\mathtt{PC}_i$ and $\mathtt{PC}'_i$ to the j-th column, respectively, for each $j \in \mathbb{Z}_c$. Then, for all j, Equation (12) holds if and only if

$$\mathsf{DPT}_{\mathtt{p}_{2,j} \circ M \circ \mathtt{p}_{1,j}} \ = \ \mathsf{DPT}_{\mathtt{p}'_{2,j} \circ M \circ \mathtt{p}'_{1,j}},$$

where M is the diffusion matrix of $\mathtt{MC}$. This, in turn, implies

$$(\mathtt{p}_{1,j}, \mathtt{p}_{2,j}) \ \overset{\mathsf{DPT}}{\sim}_M \ (\mathtt{p}'_{1,j}, \mathtt{p}'_{2,j}). \tag{13}$$

Therefore, we can use the relation $\overset{\mathsf{DPT}}{\sim}_M$ to classify the set of column-wise permutation pairs $\mathbb{S}_r \times \mathbb{S}_r$ into equivalence classes, and denote the set of representatives of all such classes by $\mathbb{P}_M$. From these, we can construct a set of representative shuffle layers under MC-DPT-equivalence relation according to Equation 12 and Equation 13; the corresponding pairs $(\mathtt{PC}_1, \mathtt{PC}_2)$ form the set $\mathbb{P}_{\mathtt{MC}}$.

Since a shuffle layer may admit multiple valid decompositions, the above construction may initially place it into several classes. However, because MC-DPT-equivalence is an equivalence relation, all classes containing the same shuffle layer can be merged into a single,

larger class, from which only one representative is sufficient. Moreover, any decomposition of a shuffle layer can be derived from that of its partition-class representative $\mathtt{P}_r$, so the decomposition of $\mathtt{P}_r$ suffices to obtain those of all shuffle layers in its partition. This completes the classification of shuffle layers under $\mathtt{MC}$-DPT-equivalence.

In cases where $A_{\mathtt{P}_r} = A_{\mathtt{Q}\circ\mathtt{P}_r\circ\mathtt{Q}^{-1}}$ for some $\mathtt{Q} \in \mathbb{P}_{\rightleftharpoons}$, redundancy may remain even after applying $\mathtt{MC}$-DPT-equivalence. We therefore further refine $\mathbb{P}_{\mathtt{MC}}$ by considering all representations of both $\mathtt{P}_r$ and $\mathtt{Q}\circ\mathtt{P}_r\circ\mathtt{Q}^{-1}$. Algorithm 1 provides the complete procedure for deriving the minimal set $\mathbb{P}$ of shuffle-layer representatives.

Given a valid structure matrix A, let $\mathtt{P}_r$ be the representative of the partition class with respect to A. The size of $\mathbb{P}_{\mathtt{MC}}$ is $|\mathbb{P}_M|^c$. Consequently, the number of elements in $\mathbb{P}$ with respect to A is bounded by

$$\frac{|\mathbb{P}_M|^c}{N_{\mathtt{P}_r}(c!)} \;\le\; |\mathbb{P}| \;\le\; |\mathbb{P}_M|^c. \tag{14}$$

Comparison with the shuffle-layer classification in [ABI+18]. The classification method in [ABI+18] (see Definition 1) is based on a notion of cipher-level equivalence, whereas our approach classifies shuffle layers according to differential-pattern equivalence. Let $\mathcal{C}(M)$ denote the set of all cell permutations in $\mathbb{S}_r$ that commute with the diffusion matrix M. Define

$$\mathcal{T}_0(M) = \{\mathtt{PC} \in \mathbb{PC} \mid \mathtt{p}_j \in \mathcal{C}(M),\ j \in \mathbb{Z}_c\},$$
$$\mathcal{T}_1(M) = \{\mathtt{Q}\circ\mathtt{PC} \mid \mathtt{PC} \in \mathcal{T}_0(M),\ \mathtt{Q} \in \mathbb{P}_{\rightleftharpoons}\},$$

where $\mathtt{p}_j$ denotes the restriction of $\mathtt{PC}$ to column j. It is easy to verify that $\mathcal{T}_1(M) \subseteq \mathcal{T}(M)$.

From Definition 1, for any $\mathtt{P}' \in \mathcal{T}_1(M)$ we have

$$\mathtt{P} \;\sim_M\; \mathtt{P}'\circ\mathtt{P}\circ\mathtt{P}'^{-1}.$$

We now show that $\mathtt{P}$ and $\mathtt{P}'\circ\mathtt{P}\circ\mathtt{P}'^{-1}$ are also equivalent under our classification.

Step 1. Suppose $\mathtt{PC} \in \mathcal{T}_0(M)$. Let $\mathtt{P}$ be the representative of the partition containing $\mathtt{P}$. Since $\mathtt{PC} \in \mathbb{PC}$, the shuffle layer $\mathtt{PC}\circ\mathtt{P}\circ\mathtt{PC}^{-1}$ belongs to the same partition as $\mathtt{P}$. Moreover, because $\mathtt{PC}$ commutes with $\mathtt{MC}$, we have

$$\mathsf{DPT}_{\mathtt{MC}} \;=\; \mathsf{DPT}_{\mathtt{PC}\circ\mathtt{MC}\circ\mathtt{PC}^{-1}},$$

and hence

$$\mathtt{P} \;\overset{\mathsf{DPT}}{\sim}_{\mathtt{MC}}\; \mathtt{PC}\circ\mathtt{P}\circ\mathtt{PC}^{-1}.$$

Step 2. Let $\mathtt{P}' \in \mathcal{T}_1(M)$. Then there exist $\mathtt{PC} \in \mathcal{T}_0(M)$ and $\mathtt{Q} \in \mathbb{P}_{\rightleftharpoons}$ such that

$$\mathtt{P}' = \mathtt{Q}\circ\mathtt{PC}.$$

Therefore,

$$\mathtt{P}'\circ\mathtt{P}\circ\mathtt{P}'^{-1} = \mathtt{Q}\circ\mathtt{PC}\circ\mathtt{P}\circ\mathtt{PC}^{-1}\circ\mathtt{Q}^{-1}.$$

Since $\mathtt{PC}\circ\mathtt{P}\circ\mathtt{PC}^{-1} \;\overset{\mathsf{DPT}}{\sim}_{\mathtt{MC}}\; \mathtt{P}$ from Step 1, conjugation by elements of $\mathbb{P}_{\rightleftharpoons}$ ensures that $\mathtt{PC}\circ\mathtt{P}\circ\mathtt{PC}^{-1} \rightleftharpoons \mathtt{P}'\circ\mathtt{P}\circ(\mathtt{P}')^{-1}$, we conclude that $\mathtt{P}$ and $\mathtt{P}'\circ\mathtt{P}\circ\mathtt{P}'^{-1}$ is equivalent under our classification.

For most diffusion matrices used in existing block ciphers, the equality $\mathcal{T}_1(M) = \mathcal{T}(M)$ holds. In this case, every equivalence class under the method of [ABI+18] is contained in a single equivalence class under our classification. Therefore, our classification is generally finer than the classification in [ABI+18]. Since differential and linear resistance are the primary security properties of concern in block-cipher design, and those are also the main focus in [ABI+18]. Our classification is naturally aligned with these objectives. In the search for shuffle layers that maximize differential resistance, the reduced number of classes provided by our method enables much more efficient traversal and facilitates the identification of high-quality candidates.

Algorithm 1: Computation of the minimal set of representatives $\mathbb{P}$ under $\approx_{\mathtt{MC}}$ and $\sim$ relations with respect to a structure matrix A

Input: The representative of the partition class with respect to A: $\mathtt{P}_r$; All $\mathtt{P}_r$'s decompositions $\{(\mathtt{PC}_{1,j}, \mathtt{PC}_{2,j})\}$ for $0 \le j < N_{\mathtt{P}_r}$; All $\mathtt{Q} \circ \mathtt{P}_r \circ \mathtt{Q}^{-1}$'s decompositions $\{(\mathtt{PC}_{1,\mathtt{Q}}, \mathtt{PC}_{2,\mathtt{Q}})\}$ for $\mathtt{Q} \in \mathbb{P}_{\rightleftharpoons}$ satisfying $A_{\mathtt{P}_r} = A_{\mathtt{Q}\circ\mathtt{P}_r\circ\mathtt{Q}^{-1}}$;

Output: The minimal set of representatives $\mathbb{P}$ with respect to A.

Step 1: Classification under $\overset{\mathsf{DPT}}{\sim}_M$.

foreach $(\mathtt{p}_1, \mathtt{p}_2) \in \mathbb{S}_r \times \mathbb{S}_r$ **do**

Derive $\mathsf{DPT}_{\mathtt{p}_2\circ M\circ\mathtt{p}_1}$ by permuting the entries of DPT_M; Use the hash table H with DPT as the key and the set of $(\mathtt{p}_1, \mathtt{p}_2)$ pairs as the value; If the current DPT is not in $\mathcal{H}$, insert a new entry and store $(\mathtt{p}_1, \mathtt{p}_2)$ as its representative in $\mathbb{P}_M$; Otherwise, add $(\mathtt{p}_1, \mathtt{p}_2)$ to the existing value set in $\mathcal{H}[\mathsf{DPT}_{\mathtt{p}_2\circ M\circ\mathtt{p}_1}]$.

Step 2: Construction of representatives candidates of $\overset{\mathsf{DPT}}{\sim}_{\mathtt{MC}}$ equivalence classes.

Generate

$$\mathbb{P}_{\mathtt{MC}} = \{(\mathtt{PC}_1, \mathtt{PC}_2) \mid (\mathtt{p}_{1,j}, \mathtt{p}_{2,j}) \in \mathbb{P}_M,\ j \in \mathbb{Z}_c\},$$

where $\mathtt{p}_{1,j}, \mathtt{p}_{2,j}$ denote the restrictions of $\mathtt{PC}_1, \mathtt{PC}_2$ to column j.

Step 3: Filtering to obtain minimal representatives.

Initialize $\mathbb{P} \leftarrow \emptyset$;

while $\mathbb{P}_{\mathtt{MC}} \neq \emptyset$ **do**

Select $(\mathtt{PC}_1, \mathtt{PC}_2) \in \mathbb{P}_{\mathtt{MC}}$ and remove it;

Insert $\mathtt{PC}_1 \circ \mathtt{P}_r \circ \mathtt{PC}_2$ into $\mathbb{P}$;

foreach *decomposition* $(\mathtt{PC}_{1,j}, \mathtt{PC}_{2,j})$ *of* $\mathtt{P}_r$ **do**

Compute

$$(\mathtt{PC}'_1, \mathtt{PC}'_2) = (\mathtt{PC}_1 \circ \mathtt{PC}_{1,j},\ \mathtt{PC}_{2,j} \circ \mathtt{PC}_2).$$

Let $\mathtt{p}'_{1,j}, \mathtt{p}'_{2,j}$ be their restrictions to column j;

Find $(\tilde{\mathtt{p}}_{1,j}, \tilde{\mathtt{p}}_{2,j}) \in \mathbb{P}_M$ such that $(\mathtt{p}'_{1,j}, \mathtt{p}'_{2,j}) \in \mathcal{H}[\mathsf{DPT}_{\tilde{\mathtt{p}}_{2,j}\circ M\circ\tilde{\mathtt{p}}_{1,j}}]$;

Extend them to $(\tilde{\mathtt{PC}}_1, \tilde{\mathtt{PC}}_2)$;

if $(\tilde{\mathtt{PC}}_1, \tilde{\mathtt{PC}}_2) \in \mathbb{P}_{\mathtt{MC}}$ **then**

Remove it from $\mathbb{P}_{\mathtt{MC}}$;

foreach *decomposition* $(\mathtt{PC}_{1,\mathtt{Q}}, \mathtt{PC}_{2,\mathtt{Q}})$ *of* $\mathtt{Q} \circ \mathtt{P}_r \circ \mathtt{Q}^{-1}$ **do**

Compute

$$(\mathtt{PC}'_1, \mathtt{PC}'_2) = (\mathtt{PC}_1^{\mathtt{Q}} \circ \mathtt{PC}_{1,\mathtt{Q}},\ \mathtt{PC}_{2,\mathtt{Q}} \circ \mathtt{PC}_2^{\mathtt{Q}}).$$

Let $\mathtt{p}'_{1,j}, \mathtt{p}'_{2,j}$ be their restrictions to column j;

Find $(\tilde{\mathtt{p}}_{1,j}, \tilde{\mathtt{p}}_{2,j}) \in \mathbb{P}_M$ such that $(\mathtt{p}'_{1,j}, \mathtt{p}'_{2,j}) \in \mathcal{H}[\mathsf{DPT}_{\tilde{\mathtt{p}}_{2,j}\circ M\circ\tilde{\mathtt{p}}_{1,j}}]$;

Extend them to $(\tilde{\mathtt{PC}}_1, \tilde{\mathtt{PC}}_2)$;

if $(\tilde{\mathtt{PC}}_1, \tilde{\mathtt{PC}}_2) \in \mathbb{P}_{\mathtt{MC}}$ **then**

Remove it from $\mathbb{P}_{\mathtt{MC}}$;

return $\mathbb{P}$.

4.3 Accelerating the Search for Optimal Shuffle Layers

In block cipher design, one typically aims for strong resistance against differential attacks. Accordingly, the shuffle layer is expected to maximize the lower bound on the number of active S-boxes over n rounds. We call a shuffle layer that achieves the maximum possible bound an *optimal shuffle layer*.

From the construction of shuffle layer representatives in the previous section and by

Equation 14, the number of representatives grows exponentially with $|\mathbb{P}_M|$, i.e., proportional to $|\mathbb{P}_M|^c$, where $|\mathbb{P}_M|$ denotes the number of $\overset{\mathtt{DPT}}{\sim}_M$ equivalence classes. For diffusion matrices with relatively few equivalence classes (e.g., PC matrices or PI matrices), enumerating all representatives is still feasible. However, for other matrices with large number of $\overset{\mathtt{DPT}}{\sim}_M$ equivalence classes the enumeration cost may become prohibitive.

To mitigate this, we exploit the partial order relation defined on DPTs. By Theorem 1, assume we construct DPT_{M_1} and DPT_{M_2} with corresponding dummy diffusion matrices M_1 and M_2 such that

$$\mathsf{DPT}_{M_1} \preceq \mathsf{DPT}_M \preceq \mathsf{DPT}_{M_2}.$$

Then, for each permutation layer, the lower bound on the number of active S-boxes for M can be estimated from the bounds for M_1 and M_2. When the number of equivalence classes under $\overset{\mathtt{DPT}}{\sim}_{M_1}$ and $\overset{\mathtt{DPT}}{\sim}_{M_2}$ is relatively small, we can first determine bounds for the cipher with diffusion matrix M, and then search for optimal shuffle layers only among promising candidates. This naturally leads to a branch-and-bound strategy:

1. Traverse all shuffle-layer representatives with respect to M_1 (resp. M_2), and compute its lower bound on the number of active S-boxes.
2. Denote by H (resp. L) the maximum lower bound obtained over all representatives with respect to M_1 (resp. M_2).
3. Initialize the global lower bound $D \leftarrow L$, and record the current best shuffle layer $\mathtt{P}'$ as the representative that achieves L under M_2.
4. For each shuffle-layer representative $\mathtt{P}$ with respect to M, note that there exists an equivalence class under M_1 (resp. M_2) containing $\mathtt{P}$. Let $h_{\mathtt{P}}$ (resp. $l_{\mathtt{P}}$) denote the corresponding lower bound under M_1 (resp. M_2). If $h_{\mathtt{P}} > D$, compute the exact lower bound d of active S-boxes under M. If $d > D$, update $D \leftarrow d$ and set $\mathtt{P}' \leftarrow \mathtt{P}$.
5. Return the optimal shuffle layer $\mathtt{P}'$ together with its corresponding lower bound D.

Complexity Analysis of Branch-and-bound Strategy Let N, N_1, and N_2 denote the numbers of shuffle-layer representatives with respect to M, M_1, and M_2, respectively. Let T, T_1, and T_2 denote the average time required to compute the lower bound on the number of active S-boxes under M, M_1, and M_2 for a single representative.

The main computational cost arises from Steps 1 and 4. The time complexity of Step 1 can be estimated as

$$N_1 \cdot T_1 + N_2 \cdot T_2.$$

The cost of Step 4 is typically dominated by computing the exact lower bound of active S-boxes for those shuffle layers satisfying $h_{\mathtt{P}} > D$. Let $N_{\text{cand}} \leq N$ denote the number of candidates that survive the filtering process and are actually evaluated in the main loop (i.e., those with $h_{\mathtt{P}} > D$ at the time of examination). In the worst case, $N_{\text{cand}} = N$. For each such candidate, computing the exact lower bound under M requires time T, so the total cost of this step is

$$N_{\text{cand}} \cdot T.$$

Combining the above contributions, the total running time with the branch-and-bound strategy can be estimated as

$$\text{Time}_{\text{BB}} = N_1 \cdot T_1 + N_2 \cdot T_2 + N_{\text{cand}} \cdot T.$$

Without applying the branch-and-bound algorithm, the total time consumption can be approximated as

$$\text{Time}_{\text{naive}} = N \cdot T.$$

Therefore, the branch-and-bound approach is effective only when

$$N_1T_1 + N_2T_2 + N_{\text{cand}}T < NT.$$

Since T, T_1, and T_2 usually do not differ significantly, the above condition can be approximately interpreted as that the sum of N_1, N_2, and N_{cand} is relatively small compared with N.

In practice, the value of N_{cand} is determined by the tightness of the bounds DPT_{M_1} and DPT_{M_2} with respect to DPT_M, as well as by the distribution of the lower bounds of active S-boxes among different shuffle layer representatives. The tighter the bounds and the broader the distribution, the smaller N_{cand} tends to be relative to N, and consequently, the more efficient the branch-and-bound algorithm becomes.

The memory complexity is at least $O(N_1 + N_2)$, which corresponds to storing the representatives and their associated results for M_1 and M_2. For M, memory usage can be controlled depending on the strategy: either only the optimal results are retained, or additional representatives and their corresponding results are selectively stored based on available memory. In the scenario where only the optimal results are kept for M, the memory overhead is about $O(N_1 + N_2)$.

Remark 1. The above branch-and-bound procedure provides a general template for identifying optimal shuffle layers with respect to differential resistance. In practice, however, cipher design often requires balancing additional criteria, such as linear resistance, diffusion properties, or algebraic degrees. In such cases, trade-offs may arise in the selection of shuffle layers. Nevertheless, the proposed branch-and-bound framework can be readily adapted by incorporating further filtering conditions to accommodate multiple design objectives, while still retaining its pruning and acceleration benefits.

5 Case Studies

To illustrate the applicability of our framework, we study the diffusion matrices used in several AES-like ciphers, namely `MIDORI`, `QARMA`, `PRINCE`, `AES` and a lightweight MDS matrix from [SS16]. For each matrix, we analyze its DPT and the induced $\overset{\mathsf{DPT}}{\sim}_M$ equivalence classes, and then construct and filter the corresponding representatives of shuffle layers. Based on this analysis, we further determine the optimal shuffle layers tailored to each matrix.

We also extend the DPT framework to heterogeneous designs in which odd and even rounds are assigned distinct diffusion matrices, using M_{midori} for odd rounds and the MDS matrix for even rounds. For these configurations, we compute lower bounds on the number of active S-boxes per round and examine how alternating matrices affect differential resistance and overall cipher performance. These experiments demonstrate both the flexibility of our DPT framework and its effectiveness in handling ciphers with heterogeneous round functions.

All experiments were conducted on a desktop PC with an Intel® Core™ i9-14900HX CPU (24 physical cores, 32 logical processors) running at 2.20 GHz. The runtime for each experimental step is reported in Table 5.

5.1 Study on the Matrix in `MIDORI`

The diffusion matrix M_{midori} used in `MIDORI` [BBI+15] is given in Equation 2. It is an almost MDS matrix, notable for its small gate size and low path delay. Consequently, it is not only adopted in the low-energy design of `MIDORI`, but also in several low-latency ciphers, including `MANTIS` [BJK+16], `Orthros` [BIL+21], and `uLBC` [LYT+25]. The corresponding $\mathsf{DPT}_{M_{\text{midori}}}$ is listed in Table 1.

It is easy to verify that M_{midori} is a PC matrix. The number of $\overset{\mathtt{DPT}}{\sim}_{M_{\text{midori}}}$ equivalence classes is $4! = 24$, and a complete set of representatives can be chosen as

$$\{(\mathtt{p}_1, \mathtt{id}) \mid \mathtt{p}_1 \in \mathbb{S}_4\}.$$

To further evaluate shuffle layers, we consider the partition class where the structure matrix is the all-one matrix. In this case, each shuffle layer maps the four cells of any column into four distinct columns, thereby satisfying the wide-trail strategy [DR02]. After classification by using Algorithm 1, we obtain $14,022 \approx 2^{13.78}$ representatives. This classification matches the result of [ABI+18], since M_{midori} itself commutes with any permutation and its DPT preserves this property but introduces no additional properties. The resulting number of representatives is sufficiently small to allow exhaustive traversal, enabling us to compute the lower bound of active S-boxes for up to 16 rounds.

Table 4 summarizes the maximum achievable lower bound on the number of active S-boxes over n rounds ($n \leq 16$). In particular, for 16 rounds we identify two classes in which the lower bound reaches 86 active S-boxes, whereas the shuffle layer $\mathtt{P}_{\text{midori}}$ employed in MIDORI achieves only 84. This observation contradicts the claim in [BBI+15] that $\mathtt{P}_{\text{midori}}$ achieves the maximum number of active S-boxes at each round. In fact, no single shuffle layer simultaneously attains the round-wise maximum for all rounds.

5.2 Study on the Matrix in QARMA

The diffusion matrix of QARMA [Ava17], denoted by M_{qarma}, is given in Equation 15. It is also an almost MDS matrix, with the same gate size and path delay as M_{midori}. The corresponding $\mathsf{DPT}_{M_{\text{qarma}}}$ is listed in Table 6.

By the definition of DPT-equivalence, we obtain $3 \times 4!$ matrix-DPT-equivalence classes of pairs $(\mathtt{p}_1, \mathtt{p}_2)$ with respect to M_{qarma}. Two pairs $(\mathtt{p}_1, \mathtt{p}_2)$ and $(\mathtt{p}'_1, \mathtt{p}'_2)$ belong to the same equivalence class if they satisfy:

- $\mathtt{p}_2 \circ \mathtt{p}_1 = \mathtt{p}'_2 \circ \mathtt{p}'_1$;
- $\mathtt{p}_1$ and $\mathtt{p}'_1$ lie in the same coset $\mathbb{G}_0 \circ \mathtt{q}_i$ for some $0 \leq i \leq 2$,

where $\mathbb{G}_0 \subset \mathbb{S}_4$ is given by

$$\begin{aligned}\mathbb{G}_0 = \{&(0,1,2,3),(0,3,2,1),(1,0,3,2),(1,2,3,0),\\ &(2,1,0,3),(2,3,0,1),(3,0,1,2),(3,2,1,0)\},\end{aligned}$$

and the coset representatives are $\mathtt{q}_0 = (0,1,2,3)$, $\mathtt{q}_1 = (0,2,1,3)$, and $\mathtt{q}_2 = (0,1,3,2)$.

Considering all shuffle layers with the all-one structure matrix, we obtain $1120698 \approx 2^{20.1}$ representatives of shuffle layers via Algorithm 1.

Reduction of search complexity. To reduce the computational complexity of finding optimal shuffle layers, we employ the branch-and-bound strategy described in Section 4.3. Specifically, we construct a dummy matrix M_q whose DPT_{M_q}, given in Table 7, satisfies

$$\mathsf{DPT}_{M_{\text{midori}}} \leq \mathsf{DPT}_{M_{\text{qarma}}} \leq \mathsf{DPT}_{M_q}.$$

Moreover, $\overset{\mathsf{DPT}}{\sim}_{M_q}$ and $\overset{\mathsf{DPT}}{\sim}_{M_{\text{midori}}}$ share the same equivalence classes, and each equivalence class of M_q corresponds to a union of 81 equivalence classes for M_{qarma}. Hence, we select the same $14,022$ representatives of shuffle layers for M_q as those for M_{midori}.

Using this setup, we first compute the lower bounds for M_q, and leverages the results obtained for M_{midori} in Section 5.1. These provide both a lower and an upper bound for the case of M_{qarma}. We then perform the search for the optimal shuffle layers of M_{qarma}. The results are summarized in Table 3.

Table 3: Bounds and actual maximum lower bound on the number of active S-boxes per round for M_{qarma}.

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Upper bound #AS	1	4	7	16	23	30	35	38	44	50	57	62	68	74	79	86
Lower bound #AS	1	4	7	16	23	28	34	37	42	50	55	61	66	70	75	82
Actual #AS	1	4	7	16	23	28	34	38	42	50	56	61	66	71	77	82

As shown in Table 3, the obtained lower and upper bounds are very tight, with several rounds achieving equality. In particular, for 16 rounds of `QARMA`, the maximum attainable minimum number of active S-boxes is 82.

From the runtimes reported in Table 5, computing the lower bounds for all shuffle-layer representatives under M_{midori} and M_q requires 18.3 CPU days and 14.1 CPU days, respectively. Since $\mathsf{DPT}_{M_{\text{qarma}}}$ is bounded by $\mathsf{DPT}_{M_{\text{midori}}}$ and DPT_{M_q}, we estimate the average time to evaluate one shuffle-layer representative under M_{qarma} as

$$T_{\text{avg}} = \frac{18.3 + 14.1}{2 \cdot 14022} \text{ CPU days} \approx 0.0011553 \text{ CPU days}.$$

With 1120698 shuffle layer representatives, the estimated cost of a naive exhaustive evaluation is

$$T_{\text{naive}} \approx T_{\text{avg}} \times 1120698 \approx 1294.74 \text{ CPU days}.$$

By contrast, our branch-and-bound procedure takes

$$T_{\text{BB}} \approx (284 + 18.3 + 14.1) \text{ CPU days} \approx 316.4 \text{ days}.$$

Hence the branch-and-bound algorithm reduces the running time by

$$T_{\text{naive}} - T_{\text{BB}} \approx 978.34 \text{ CPU days},$$

which corresponds to a speed-up factor of

$$\frac{T_{\text{naive}}}{T_{\text{BB}}} \approx 4.09.$$

5.3 Study on the Matrices in `PRINCE`

`PRINCE` [BCG+12] employs two diffusion matrices, denoted by M_{prince1} and M_{prince2}, as given in Equation 16. Both M_{prince1} and M_{prince2} are almost MDS matrices, with the same gate size and path delay as M_{midori}. The corresponding $\mathsf{DPT}_{M_{\text{prince1}}}$ and $\mathsf{DPT}_{M_{\text{prince2}}}$ are identical, and are listed in Table 8. In addition, both M_{prince1} and M_{prince2} are RC matrices.

The number of $\overset{\mathsf{DPT}}{\sim}_{M_{\text{prince1}}}$-equivalence classes is 9, and the set of representatives can be written as

$$\{(\mathtt{p}_1, \mathtt{p}_2) \mid \mathtt{p}_1, \mathtt{p}_2 \in \mathbb{G}_1\}, \quad \mathbb{G}_1 = \{(0,1,2,3),\ (0,2,1,3),\ (0,1,3,2)\}.$$

Considering all shuffle layers with the all-one structure matrix, we obtain 288 representatives via Algorithm 1. Remarkably, the lower bound on the number of active S-boxes is identical across all representatives, with the 16-round bound reaching 64. Table 4 summarizes the bounds for n rounds, where $n \leq 16$.

5.4 Study on the MDS Matrix in [SS16]

The matrix presented in [SS16], denoted as M_t and shown in Equation 17, is a lightweight MDS matrix with a branch number of 5. Compared to M_{midori}, M_t has a larger gate size and a longer path delay. The corresponding DPT_{M_t} is listed in Table 9.

It can be verified that M_t is a PI matrix. Consequently, the number of $\overset{\mathsf{DPT}}{\sim}_{M_t}$ equivalence classes is 1. Therefore, for shuffle layers with an all-one structure matrix, there is only a single representative, which we choose to be the identity permutation `ID`.

The resulting bounds on the minimum number of active S-boxes are summarized in Table 4, where the 16-round minimum reaches 100.

5.5 Study on the Matrix in `AES`

The matrix used in `AES` [DR02], denoted by M_{aes} and shown in Equation 18, is an MDS matrix with a branch number of 5. Its corresponding $\mathsf{DPT}_{M_{\text{aes}}}$, listed in Table 9, coincides with the DPT of M_t. Consequently, M_{aes} is also a PI matrix, and the number of $\overset{\mathsf{DPT}}{\sim}_{M_{\text{aes}}}$ equivalence classes is 1. The resulting lower bounds on the number of active S-boxes are therefore identical to those obtained for M_t, and are summarized in Table 4, where the 16-round minimum reaches 100.

Table 4: Comparison of the maximum low bound of active S-boxes number per round with that from the original shuffle layer

Matrix	B_d	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Ref
M_{midori}	4	1	4	7	16	23	30	35	38	44	50	57	62	68	74	79	86	Sec. 5.1
		1	4	7	16	23	30	35	38	41	50	57	62	67	72	75	84	[BBI+15]
M_{qarma}	4	1	4	7	16	23	28	34	38	42	50	56	61	66	71	77	82	Sec. 5.2
		1	4	7	16	23	28	34	37	41	50	55	61	66	71	75	82	[Ava17]
M_{prince1} (M_{prince2})	4	1	4	7	16	17	20	23	32	33	36	39	48	49	52	55	64	Sec. 5.3
		1	4	7	16	17	20	23	32	33	36	39	48	49	52	55	64	[BCG+12]
M_t	5	1	5	9	25	26	30	34	50	51	55	59	75	76	80	84	100	Sec. 5.4
M_{aes}	5	1	5	9	25	26	30	34	50	51	55	59	75	76	80	84	100	Sec. 5.5
M_{midori} + M_t	4 5	1	4	8	20	24	32	34	40	48	54	61	64	72	77	86	92	Sec. 5.6

5.6 Extend Application to Heterogeneous Diffusion Matrices Designs

In this subsection, we extend the applicability of the DPT framework to heterogeneous round-function constructions, a design approach that has appeared in several works [SS04, SP04, HKPT24]. Such constructions employ different round functions across rounds, allowing designers to combine multiple linear or nonlinear components in order to enhance certain security properties or implementation characteristics. Heterogeneous round designs are well suited for fully unrolled hardware implementations, such as memory encryption and pointer authentication, where their flexibility enables additional reductions in critical-path delay for ultra-low-latency operation.

Here, we focus on a concrete construction in which two distinct diffusion matrices are applied alternately in odd and even rounds, while a single shuffle layer is used uniformly across all rounds. Specifically, we choose M_{midori} from `MIDORI` for odd rounds, as it is one of the strongest branch-4 matrices in terms of differential resistance in the above evaluation, and the MDS matrix M_t for even rounds. Since M_{midori} belongs to the class of PC matrices and M_t to the class of PI matrices (which implies PC), any permutation pair

within a $\overset{\mathsf{DPT}}{\sim} M_{\text{midori}}$ equivalence class also belongs to the same $\overset{\mathsf{DPT}}{\sim}_{M_t}$ equivalence class. Consequently, for shuffle layers with an all-one structure matrix, it suffices to consider the 14,022 representatives identified in Section 5.1, making this case study both analytically feasible and representative.

We then compute the maximum lower bound on the number of active S-boxes for each round. The results are summarized in Table 4. Notably, although M_{midori} has smaller delay and area than M_t, the lower bound on active S-boxes for rounds 6, 11, and 15 surpasses that achieved using M_t alone. This demonstrates the potential of alternating diffusion matrices across different rounds to implement lower-latency cipher designs, and also shows that our methodology can effectively handle ciphers employing multiple diffusion matrices.

Table 5: Runtime (CPU time) for different steps of the DPT analysis on various diffusion matrices.

Matrix	Compute DPT_M	Classify $\overset{\mathsf{DPT}}{\sim}_M$	Compute shuffle representatives	Find optimal permutation layer	Ref
M_{midori}	< 1 s	< 1 s	37 s	18.3 d	Sec. 5.1
M_{qarma}	< 1 s	< 1 s	0.86 h	284.0 d	Sec. 5.2
M_q	—	< 1 s	—	14.1 d	Sec. 5.2
M_{prince1} (M_{prince2})	< 1 s	< 1 s	1 s	0.9 h	Sec. 5.3
M_t	< 1 s	< 1 s	< 1 s	13 s	Sec. 5.4
M_{AES}	19.6 h	—	—	—	Sec. 5.5
$M_{\text{midori}} + M_t$	—	—	—	15.8 d	Sec. 5.6

5.7 Discussion

Comparison with the shuffle-layer classification in [ABI+18]. In all diffusion matrices considered in our study, the relation

$$\mathcal{T}_1(M) = \mathcal{T}(M)$$

holds. Consequently, in these cases, the number of equivalence classes obtained by our classification never exceeds that obtained by the method in [ABI+18].

This contrast becomes particularly evident for the MDS matrix M_t from [SS16]. For shuffle layers whose structure matrix is the all-one matrix, our DPT-based classification yields only a single equivalence class. In contrast, under the method of [ABI+18], since the identity is the only permutation that commutes with M_t, the set of shuffle layers commuting with the MC operation defined by M_t is exactly $\mathbb{P}_{\rightleftharpoons}$. By Definition 1, this implies that the number of equivalence classes for shuffle layers with an all-one structure matrix is at least

$$\frac{(4!)^8}{4!} = (4!)^7.$$

This substantial difference further confirms the effectiveness of our classification approach in identifying shuffle layers with optimal differential resistance.

Discussion on the Design of Linear Layers. With the above case studies, we now discuss the implications of design choices for diffusion matrices and shuffle layers. Our main observations are as follows:

- Even when diffusion matrices share the same branch number, their differential resistance can differ significantly. For instance, M_{midori}, M_{qarma} and M_{prince1} have identical branch numbers, but their differential pattern transitions satisfy $\mathsf{DPT}_{M_{\text{midori}}} \leq \mathsf{DPT}_{M_{\text{qarma}}} \leq \mathsf{DPT}_{M_{\text{prince1}}}$. This ordering directly implies that M_{midori} enables the construction of ciphers with potentially stronger differential resistance.

- A larger branch number does not necessarily guarantee superior differential resistance. For example, the maximum number of active S-boxes over 7 rounds is 35 for M_{midori}, but only 34 for M_t, despite $B_d(M_{\text{midori}}) < B_d(M_t)$.

- The choice of shuffle layer has a profound impact on resistance to differential attacks, as is evident from the distribution of the lower bound on the number of active S-boxes across all representations shown in Figure 1. Among all 14022 representatives, only two representatives achieve 86 active S-boxes over 16 rounds, whereas 2649 representatives reach only 64 active S-boxes. This highlights the necessity of classifying shuffle layers and systematically searching for those that maximize differential resistance.

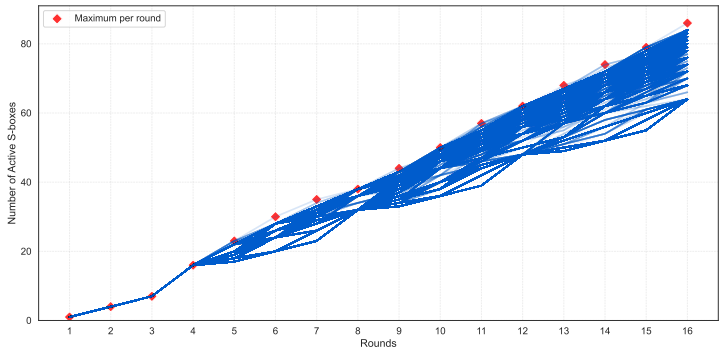


Figure 1: Distribution of the lower bound on the number of active S-boxes for all shuffle-layer representatives in a 16-round `AES`-like cipher with M_{midori}.

6 Conclusion

In this paper, we introduced a new cryptographic notion, the *differential pattern transition* (DPT), for analyzing the differential properties of diffusion matrices. The DPT enables a finer-grained evaluation of differential strength across multiple rounds, capturing distinctions not reflected by the classical branch number. Building on this notion, we developed a systematic framework for classifying and filtering shuffle layers, and proposed a branch-and-bound strategy to efficiently identify optimal permutation layers even when the number of representatives is large.

We applied our framework to several existing ciphers, including `MIDORI`, `PRINCE`, `QARMA`, `AES`, and a lightweight MDS matrix from [SS16], demonstrating how the DPT framework can guide both the selection of diffusion matrices and the shuffle layers. We further extended its application to heterogeneous constructions in which different diffusion matrices are used in odd and even rounds, showing both the effectiveness of our framework in such settings and the potential of these designs to achieve higher differential resistance while reducing delay or resource usage.

Overall, our work provides both theoretical insights and practical tools for designing secure and efficient `AES`-like ciphers, and highlights new opportunities for multi-round optimization in lightweight and low-latency cryptographic constructions.

Acknowledgments

The authors acknowledge the reviewers for their insightful comments and detailed suggestions which helped improve this manuscript significantly. This work is supported by the National Key Research and Development Program of China (2024YFB4504800), the National Cryptologic Science Foundation of China (2025NCSF01012), the National Natural Science Foundation of China (62532013, 62202444), the Fellowship Funds from China Postdoctoral Science Foundation (2025M771675), the Fundamental Research Funds for the Central Universities and the Scientific Research Innovation Capability Support Project for Young Faculty.

References

[ABI+18] Gianira Alfarano, Christof Beierle, Takanori Isobe, Stefan Kölbl, and Gregor Leander. Shiftrows alternatives for AES-like ciphers and optimal cell permutations for Midori and Skinny. *IACR Trans. Symm. Cryptol.*, 2018(2):20–47, 2018.

[Ava17] Roberto Avanzi. The QARMA block cipher family. *IACR Trans. Symm. Cryptol.*, 2017(1):4–44, 2017.

[BBI+15] Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni. Midori: A block cipher for low energy. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part II*, volume 9453 of *LNCS*, pages 411–436. Springer, Heidelberg, November / December 2015.

[BCG+12] Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knežević, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçin. PRINCE - A low-latency block cipher for pervasive computing applications - extended abstract. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 208–225. Springer, Heidelberg, December 2012.

[BIL+21] Subhadeep Banik, Takanori Isobe, Fukang Liu, Kazuhiko Minematsu, and Kosei Sakamoto. Orthros: A low-latency PRF. *IACR Trans. Symm. Cryptol.*, 2021(1):37–77, 2021.

[BJK+16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 123–153. Springer, Heidelberg, August 2016.

[BS91] Eli Biham and Adi Shamir. Differential cryptanalysis of DES-like cryptosystems. In Alfred J. Menezes and Scott A. Vanstone, editors, *CRYPTO'90*, volume 537 of *LNCS*, pages 2–21. Springer, Heidelberg, August 1991.

[Dae95] Joan Daemen. *Cipher and hash function design, strategies based on linear and differential cryptanalysis, PhD Thesis.* K.U.Leuven, 1995. http://jda.noekeon.org/.

[DKR97] Joan Daemen, Lars Knudsen, and Vincent Rijmen. The block cipher Square. In Eli Biham, editor, *Fast Software Encryption*, pages 149–165, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

[DR02] Joan Daemen and Vincent Rijmen. *The design of Rijndael*, volume 2. Springer, 2002.

[EHDRM23] Solane El Hirch, Joan Daemen, Raghvendra Rohit, and Rusydi H Makarim. Twin column parity mixers and Gaston: A new mixing layer and permutation. In *Annual International Cryptology Conference*, pages 475–506. Springer, 2023.

[GPP11] Jian Guo, Thomas Peyrin, and Axel Poschmann. The PHOTON family of lightweight hash functions. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 222–239. Springer, Heidelberg, August 2011.

[GPPR11] Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J. B. Robshaw. The LED block cipher. In Bart Preneel and Tsuyoshi Takagi, editors, *CHES 2011*, volume 6917 of *LNCS*, pages 326–341. Springer, Heidelberg, September / October 2011.

[HKPT24] Kai Hu, Mustafa Khairallah, Thomas Peyrin, and Quan Quan Tan. uKNIT: Breaking round-alignment for cipher design – featuring uKNIT-BC, an ultra low-latency block cipher. Cryptology ePrint Archive, Paper 2024/1962, 2024.

[KW02] Lars Knudsen and David Wagner. Integral cryptanalysis. In Joan Daemen and Vincent Rijmen, editors, *Fast Software Encryption*, pages 112–127, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[LK06] Chae Hoon Lim and Tymur Korkishko. mCrypton - A lightweight block cipher for security of low-cost RFID tags and sensors. In Jooseok Song, Taekyoung Kwon, and Moti Yung, editors, *WISA 05*, volume 3786 of *LNCS*, pages 243–258. Springer, Heidelberg, August 2006.

[LRL+24] Hao Lei, Raghvendra Rohit, Guoxiao Liu, Jiahui He, Mohamed Rachidi, Keting Jia, Kai Hu, and Meiqin Wang. Symmetric twin column parity mixers and their applications. *IACR Transactions on Symmetric Cryptology*, 2024(4):1–37, 2024.

[LS16] Meicheng Liu and Siang Meng Sim. Lightweight MDS generalized circulant matrices. In Thomas Peyrin, editor, *FSE 2016*, volume 9783 of *LNCS*, pages 101–120. Springer, Heidelberg, March 2016.

[LSL+19] Shun Li, Siwei Sun, Chaoyun Li, Zihao Wei, and Lei Hu. Constructing low-latency involutory MDS matrices with lightweight circuits. *IACR Trans. Symm. Cryptol.*, 2019(1):84–117, 2019.

[LSS+19] Shun Li, Siwei Sun, Danping Shi, Chaoyun Li, and Lei Hu. Lightweight iterative MDS matrices: How small can we go? *IACR Trans. Symm. Cryptol.*, 2019(4):147–170, 2019.

[LW17] Chaoyun Li and Qingju Wang. Design of lightweight linear diffusion layers from near-MDS matrices. *IACR Trans. Symm. Cryptol.*, 2017(1):129–155, 2017.

[LYT+25] Guoxiao Liu, Qingyuan Yu, Liyuan Tang, Shihe Ma, Congming Wei, Keting Jia, Lingyue Qin, Xiaoyang Dong, and Yantian Shen. Ultra low-latency block cipher uLBC. *IACR Communications in Cryptology*, 1(4), 2025.

[Mat94] Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In Tor Helleseth, editor, *EUROCRYPT'93*, volume 765 of *LNCS*, pages 386–397. Springer, Heidelberg, May 1994.

[MWGP12] Nicky Mouha, Qingju Wang, Dawu Gu, and Bart Preneel. Differential and linear cryptanalysis using mixed-integer linear programming. In Chuan-Kun Wu, Moti Yung, and Dongdai Lin, editors, *Information Security and Cryptology*, pages 57–76, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[Sha49] Claude E. Shannon. Communication theory of secrecy systems. *Bell Systems Technical Journal*, 28(4):656–715, 1949.

[SHW+14] Siwei Sun, Lei Hu, Peng Wang, Kexin Qiao, Xiaoshuang Ma, and Ling Song. Automatic security evaluation and (related-key) differential characteristic search: Application to SIMON, PRESENT, LBlock, DES(L) and other bit-oriented block ciphers. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 158–178. Springer, Heidelberg, December 2014.

[SKOP15] Siang Meng Sim, Khoongming Khoo, Frédérique E. Oggier, and Thomas Peyrin. Lightweight MDS involution matrices. In Gregor Leander, editor, *FSE 2015*, volume 9054 of *LNCS*, pages 471–493. Springer, Heidelberg, March 2015.

[SP04] Taizo Shirai and Bart Preneel. On feistel ciphers using optimal diffusion mappings across multiple rounds. In *Advances in Cryptology - ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5-9, 2004, Proceedings*, volume 3329 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2004.

[SS04] Taizo Shirai and Kyoji Shibutani. Improving immunity of feistel ciphers against differential cryptanalysis by using multiple MDS matrices. In Bimal Roy and Willi Meier, editors, *Fast Software Encryption*, pages 260–278, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[SS16] Sumanta Sarkar and Habeeb Syed. Lightweight diffusion layer: Importance of Toeplitz matrices. *IACR Trans. Symm. Cryptol.*, 2016(1):95–113, 2016. https://tosc.iacr.org/index.php/ToSC/article/view/537.

[Tro92] William T Trotter. *Combinatorics and partially ordered sets*. 1992.

[YZW21] Yumeng Yang, Xiangyong Zeng, and Shi Wang. Construction of lightweight involutory MDS matrices. *Designs, Codes and Cryptography*, 89(7):1453–1483, 2021.

[ZWFS09] Wentao Zhang, Wenling Wu, Dengguo Feng, and Bozhan Su. Some new observations on the SMS4 block cipher in the Chinese WAPI standard. In Feng Bao, Hui Li, and Guilin Wang, editors, *Information Security Practice and Experience*, pages 324–335, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

A Specification of the Matrices

The Matrix in `QARMA`. The matrix in `QARMA` is:

$$M_{\text{qarma}} = \begin{pmatrix} 0 & \rho & \rho^2 & \rho \\ \rho & 0 & \rho & \rho^2 \\ \rho^2 & \rho & 0 & \rho \\ \rho & \rho^2 & \rho & 0 \end{pmatrix}, \tag{15}$$

where ρ is the image of X in the quotient ring $\mathbb{F}_2[X]/(X^4+1)$.

The Matrices in PRINCE. The 16×16 matrices in PRINCE are:

$$M_{\text{prince1}} = \begin{pmatrix} M_0 & M_1 & M_2 & M_3 \\ M_1 & M_2 & M_3 & M_0 \\ M_2 & M_3 & M_0 & M_1 \\ M_3 & M_0 & M_1 & M_2 \end{pmatrix}, \quad M_{\text{prince2}} = \begin{pmatrix} M_1 & M_2 & M_3 & M_0 \\ M_2 & M_3 & M_0 & M_1 \\ M_3 & M_0 & M_1 & M_2 \\ M_0 & M_1 & M_2 & M_3 \end{pmatrix}, \tag{16}$$

where $M_0, \ldots, M_3$ are 4×4 submatrices given by

$$M_0 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad M_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$M_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad M_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

The Matrix in [SS16]. The MDS matrix in [SS16] over $\mathbb{F}_2^4$ is:

$$M_t = \begin{pmatrix} 1 & 1 & \alpha & \alpha^{-1} \\ \alpha^{-2} & 1 & 1 & \alpha \\ 1 & \alpha^{-2} & 1 & 1 \\ \alpha^{-1} & 1 & \alpha^{-2} & 1 \end{pmatrix}, \tag{17}$$

where α is a root of $X^4 + X^3 + 1$, generating $\mathbb{F}_2^4$.

The Matrix in AES. The MDS matrix in AES over $\mathbb{F}_2^8$ is:

$$M_{\text{aes}} = \begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix}, \tag{18}$$

where the entries are elements of $\mathbb{F}_{2^8}$ and multiplication is performed in $\mathbb{F}_{2^8}$ defined by the irreducible polynomial

$$p(x) = x^8 + x^4 + x^3 + x + 1.$$

B The DPT of matrices

See Table 6, 7, 8, 9.

C Supplementary Complexity Analysis

Computational Complexity of the DPT. Given a diffusion matrix $M : (\mathbb{F}_2^s)^m \to (\mathbb{F}_2^s)^m$, the computational complexity of deriving DPT_M is approximately

$$2^{sm} \cdot C_0,$$

where C_0 denotes the cost of multiplying an $sm \times sm$ binary matrix by an sm-dimensional binary vector. In our implementation, the matrix is encoded as sm integers and the vector as a single integer, allowing the matrix-vector multiplication to be performed efficiently using bitwise AND operations, computing the population count (i.e., the bit-level Hamming weight) of each intermediate result, and reducing each count modulo 2. Therefore, C_0 can be approximated as the cost of sm bitwise AND operations, sm population count computations, and sm modulo 2 reductions.

Table 6: DPT of M_{qarma}.

input pattern	output pattern
0000	0000
0001	1110
0010	1101
0011	0011 0111 1011 1111
0100	1011
0101	0101 1111
0110	0110 0111 1110 1111
0111	0011 0110 0111 1000 1011 1101 1110 1111
1000	0111
1001	1001 1011 1101 1111
1010	1010 1111
1011	0011 0100 0111 1001 1011 1101 1110 1111
1100	1100 1101 1110 1111
1101	0010 0111 1001 1011 1100 1101 1110 1111
1110	0001 0110 0111 1011 1100 1101 1110 1111
1111	0011 0101 0110 0111 1001 1010 1011 1100 1101 1110 1111

Complexity Analysis of Algorithm 1. Algorithm 1 comprises three phases. In Step 1, all permutation pairs $(\mathtt{p}_1, \mathtt{p}_2) \in \mathbb{S}_r \times \mathbb{S}_r$ are classified according to the equivalence relation $\overset{\mathsf{DPT}}{\sim}_M$, and a hash table $\mathcal{H}$ is constructed to record this classification. Given a precomputed DPT_M, each $\mathsf{DPT}_{\mathtt{p}_2 \circ M \circ \mathtt{p}_1}$ is derived by permuting the entries of DPT_M. The classification then proceeds by inserting and querying these DPTs in the hash table, which maps each DPT to its corresponding permutation pairs $(\mathtt{p}_1, \mathtt{p}_2)$. Hence, the overall computational cost of Step 1 is

$$2^{rs} \cdot C_0 + (r!)^2 \cdot C_1,$$

where C_0 represents the cost of a single matrix-vector multiplication as defined above and C_1 denotes the amortized cost of permuting the entries and hash insertion/lookup per permutation pair.

In Step 2, the candidate set $\mathbb{P}_{\mathtt{MC}}$ is constructed by iterating over all elements of $\mathbb{P}_M$ for each column. Hence, generating all candidates requires

$$|\mathbb{P}_M|^c \cdot C_2$$

time, where c is the number of columns and C_2 denotes the average cost of constructing one candidate.

Step 3 filters $\mathbb{P}_{\mathtt{MC}}$ by iterating over the decompositions of $\mathtt{P}_r$ and of $\mathtt{Q} \circ \mathtt{P}_r \circ \mathtt{Q}^{-1}$. Let $N_{\mathtt{P}_r}$ and N_Q denote the respective numbers of decompositions; processing each candidate then requires

$$O\big((N_{\mathtt{P}_r} + N_Q) \cdot c \cdot C_3 + C_4\big)$$

work, where C_3 represents the average cost of searching for the representative of a permutation pair, and C_4 denotes the cost of locating a candidate in $\mathbb{P}_{\mathtt{MC}}$.

Table 7: DPT of M_q.

input pattern	output pattern
0000	0000
0001	1110
0010	1101
0011	0011 0111 1011 1111
0100	1011
0101	0101 0111 1101 1111
0110	0110 0111 1110 1111
0111	0011 0101 0110 0111 1000 1011 1101 1110 1111
1000	0111
1001	1001 1011 1101 1111
1010	1010 1011 1110 1111
1011	0011 0100 0111 1001 1010 1011 1101 1110 1111
1100	1100 1101 1110 1111
1101	0010 0101 0111 1001 1011 1100 1101 1110 1111
1110	0001 0110 0111 1010 1011 1100 1101 1110 1111
1111	0011 0101 0110 0111 1001 1010 1011 1100 1101 1110 1111

Combining all three steps, the overall computational complexity of the algorithm can be bounded as

$$\text{Time} = 2^{rs} \cdot C_0 + (r!)^2 \cdot C_1 + |\mathbb{P}_M|^c \cdot \big(C_4 + (N_{\mathtt{P}_r} + N_Q) \cdot c \cdot C_3 + C_2\big).$$

When $rs \leq 32$, the first two terms, $2^{rs} \cdot C_0 + (r!)^2 \cdot C_1$, are computationally feasible on a standard computer. Hence, under this condition, the practical feasibility of Algorithm 1 primarily depends on the magnitudes of $|\mathbb{P}_M|$ and c.

The memory usage is dominated by the storage of the candidate set $\mathbb{P}_{\mathtt{MC}}$ and $\mathbb{P}$. Thus, the overall memory complexity can be estimated as

$$\text{Memory} = \mathcal{O}(|\mathbb{P}_M|^c).$$

In practice, it is not necessary to load the entire set $\mathbb{P}_{\mathtt{MC}}$ into memory at once. Instead, a set can be maintained to record the candidates that need to be removed from $\mathbb{P}_{\mathtt{MC}}$, thereby reducing the peak memory consumption.

Table 8: DPT of M_{prince1} (M_{prince2}).

input pattern	output pattern
0000	0000
0001	0111 1011 1101 1110 1111
0010	0111 1011 1101 1110 1111
0011	0011 0110 0111 1001 1011 1100 1101 1110 1111
0100	0111 1011 1101 1110 1111
0101	0101 0111 1010 1011 1101 1110 1111
0110	0011 0110 0111 1001 1011 1100 1101 1110 1111
0111	0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
1000	0111 1011 1101 1110 1111
1001	0011 0110 0111 1001 1011 1100 1101 1110 1111
1010	0101 0111 1010 1011 1101 1110 1111
1011	0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
1100	0011 0110 0111 1001 1011 1100 1101 1110 1111
1101	0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
1110	0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
1111	0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

Table 9: DPT of M_t / M_{aes}.

input pattern	output pattern
0000	0000
0001	1111
0010	1111
0011	0111 1011 1101 1110 1111
0100	1111
0101	0111 1011 1101 1110 1111
0110	0111 1011 1101 1110 1111
0111	0011 0101 0110 0111 1001 1010 1011 1100 1101 1110 1111
1000	1111
1001	0111 1011 1101 1110 1111
1010	0111 1011 1101 1110 1111
1011	0011 0101 0110 0111 1001 1010 1011 1100 1101 1110 1111
1100	0111 1011 1101 1110 1111
1101	0011 0101 0110 0111 1001 1010 1011 1100 1101 1110 1111
1110	0011 0101 0110 0111 1001 1010 1011 1100 1101 1110 1111
1111	0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111