Integral Attack on Reduced-Round Kalyna

Nitish Kumar, Ranit Dutta, Bimal Mandal

Abstract

We study integral cryptanalysis of the Ukrainian block cipher Kalyna and focus on constructing reduced-round distinguishers and key-recovery attacks with low data, time, and memory complexities. Although Kalyna has an SPN-type round structure, its pre-whitening and post-whitening layers use column-wise addition modulo 2^{64}, which makes the propagation of integral properties more delicate than in XOR-only designs. By combining carefully chosen input multisets with backward extension through inverse round transformations, we obtain integral distinguishers for Kalyna-128, Kalyna-256, and Kalyna-512 in the standard setting, under weak-key assumptions, and in variants without pre-whitening. These distinguishers require as few as 2^8 or 2^{16} chosen texts, substantially improving the data complexity of previously reported public integral results on Kalyna. We further extend them to key-recovery attacks on reduced-round Kalyna by partial decryption and balancedness tests on suitable intermediate states. For example, we obtain a 5-round key-recovery attack on Kalyna-128/128 with data complexity 2^9 chosen plaintexts, time complexity 2^{74} encryptions, and negligible memory. To the best of our knowledge, this is the first work to provide integral cryptanalysis of Kalyna-256/256 and Kalyna-512/512. Overall, our results give a unified integral analysis of Kalyna across its standard block sizes and clarify the effect of modular whitening on reduced-round distinguishers and key-recovery attacks.

Keywords: Block cipher, Kalyna, Integral distinguisher, Key-recovery attack.

1. Introduction

The origin of integral cryptanalysis can be traced back to the study of higher-order derivatives of Boolean and vectorial Boolean functions. In 1994, Lai introduced the notion of higher-order derivatives as a natural generalization of classical differential cryptanalysis and discussed their cryptographic relevance [1]. Soon after, Knudsen showed how higher-order and truncated differentials can be used to attack block ciphers that are resistant to ordinary differential attacks [2]. These ideas motivated the analysis of chosen sets of plaintexts rather than individual plaintext pairs.

A major step in this direction was the square attack. The design and analysis of the block cipher square demonstrated that, for carefully chosen plaintext structures, the sum of certain intermediate or output bytes becomes balanced after a number of rounds [3]. This viewpoint was later formalized by Knudsen and Wagner under the name *integral cryptanalysis*, where the attacker studies how properties such as *All*, *Constant*, and *Balanced* propagate through the cipher. They also described integral cryptanalysis as a dual view of differential cryptanalysis and showed that it is particularly suitable for ciphers built from bijective components [4]. Related variants such as the saturation attack further extended the usefulness of this approach to additional cipher structures [5].

Although the original integral method was highly insightful, its manual application becomes difficult for ciphers with large states, complicated linear layers, or mixed arithmetic operations.

[1]Department of Mathematics, Indian Institute of Technology Jodhpur, Karwar–342030, India
Email: p23ma0010@iitj.ac.in, p22ma207@iitj.ac.in, bimalmandal@iitj.ac.in

A major breakthrough came from Todo's division property, introduced as a generalized integral property that enables a more systematic description of the propagation of integral characteristics through nonlinear and linear components [6]. This approach led to powerful results, including the first full-round integral attack on MISTY1 [8]. To improve precision further, Todo and Morii [7] introduced the bit-based division property, and Xiang *et al.* [9] showed how the search for integral distinguishers can be automated through mixed-integer linear programming (MILP), making the technique practical for a wide range of modern block ciphers.

In this work, we focus on the block cipher *Kalyna* [14], the Ukrainian national encryption standard. Kalyna was selected in the Ukrainian public cryptographic competition and later standardized as DSTU 7624:2014. It follows a Rijndael-like substitution–permutation network design, but incorporates several distinctive features, including a larger MDS matrix, four different 8-bit S-boxes, and pre- and post-whitening by addition modulo 2^{64}. Kalyna supports block sizes of 128, 256, and 512 bits, while the key size can be either equal to or twice the block size.These design choices make Kalyna an attractive and nontrivial target for integral analysis: the strong diffusion helps spread activity rapidly, while the modular additions and large linear layer make the propagation of integral properties more subtle than in standard XOR-only SPN designs.

The existing literature on Kalyna cryptanalysis has so far focused mainly on reduced-round attacks of other types, especially meet-in-the-middle style techniques. For example, AlTawy *et al.* [11] presented reduced-round meet-in-the-middle attacks on Kalyna-$b/2b$, and subsequent work extended key-recovery results on reduced-round variants such as Kalyna-128/256 and Kalyna-256/512 [12]. In contrast, integral cryptanalysis of Kalyna appears to be much less explored. In particular, ElSheikh and Youssef [13] considered Kalyna in their MILP-based study of block ciphers with large linear layers and reported, to the best of current public knowledge, a 4-round integral distinguisher for Kalyna-128 with data complexity 2^{96}, together with a 4-round integral distinguisher without pre-whitening with data complexity 2^{64}. Hence, the currently available integral results for Kalyna seem to originate primarily from a general automated MILP framework. In the present work, we apply the traditional integral distinguisher methodology by tracking the propagation of carefully chosen multisets directly through the round structure of Kalyna, while also using backward propagation to extend the obtained round distinguishers. For example, in the case of Kalyna-128, this approach leads to a family of integral distinguishers based on a multiset of size 2^8. In particular, we obtain a 2-round integral distinguisher in the presence of pre-whitening, a 3-round weak-key integral distinguisher, and a 4-round integral distinguisher when the pre-whitening operation is omitted. This approach provides a more direct cipher-oriented understanding of how the substitution layer, byte permutation, linear diffusion, and modular whitening influence the existence and extension of integral distinguishers.

Our goal is to analyze the propagation of integral properties through the round function of Kalyna, derive meaningful distinguishers for reduced-round variants, and examine how these properties may be exploited in key recovery. Beyond the specific results for Kalyna, such a study also helps clarify how modern integral techniques interact with SPN ciphers that combine AES-like diffusion with modular arithmetic components. In this paper, we study each variant of Kalyna by starting from a carefully chosen input set. For each version, we represent the internal state as a byte matrix of the corresponding size and select a structured set of plaintexts in which certain byte positions take all possible values, while the remaining bytes are fixed to constants. We then examine how this input structure propagates through the round function and determine, for each byte of the intermediate state, whether it satisfies the integral property *All*, *Balanced*, *Constant*, or *Unknown*. Such small and well-structured input sets are particularly useful for understanding the diffusion behavior of the different Kalyna variants, for deriving integral distinguishers on reduced-round versions, and for developing key-recovery attacks by exploiting these distinguishers.

Contributions and organizations

This paper presents a systematic integral analysis of all standard variants of Kalyna, namely Kalyna-128, Kalyna-256, and Kalyna-512. Representing the internal state as a byte matrix of the appropriate dimension, we study how carefully chosen plaintext multisets propagate through the substitution layer, the ShiftRows operation, the MDS-based diffusion layer, and the modular pre-whitening transformation. In addition, we use backward propagation to extend the obtained round distinguishers and thereby derive longer integral properties. Based on these distinguishers, we further describe corresponding key-recovery attacks, whose attacked rounds and main complexities are summarized in Table 1. Our main contributions are as follows.

- For Kalyna-128, we derive a family of integral distinguishers based on a multiset of size 2^8. In particular, we obtain a 2-round integral distinguisher in the presence of pre-whitening, a 3-round weak-key integral distinguisher, and a 4-round integral distinguisher when the pre-whitening operation is omitted.
- For Kalyna-256, we derive a 2-round integral distinguisher with data complexity 2^8, together with 3-round and 4-round weak-key integral distinguishers with data complexity 2^{16}.
- For Kalyna-512, we obtain a 3-round integral distinguisher with data complexity 2^8 in the presence of pre-whitening, a 4-round weak-key integral distinguisher, and a 5-round integral distinguisher when the pre-whitening operation is not taken into account.
- We further show that these distinguishers naturally extend to key-recovery attacks on reduced-round Kalyna. In particular, the 3-round weak-key integral distinguishers for Kalyna-128 and Kalyna-256 are extended to 4-round key-recovery attacks by partial decryption of the last round, and also admit direct 5-round extensions for Kalyna-128/256 and Kalyna-256/512, respectively. In addition, the distinguisher without pre-whitening for Kalyna-512 can be extended to a 6-round key-recovery attack.
- Altogether, our results provide a unified picture of how integral properties propagate in Kalyna under the standard setting, under weak-key assumptions, and in reduced variants without pre-whitening. They also show that, because Kalyna performs round-key addition column-wise modulo 2^{64}, the natural recovery unit in the key-recovery phase is a full 64-bit column subkey rather than an individual key byte.

The remainder of this paper is organized as follows. In Section 2, we introduce the notation, recall the basic integral properties used in our analysis, and describe the round structure of Kalyna. Sections 3, 4, and 5 are devoted to the integral analysis of Kalyna-128, Kalyna-256, and Kalyna-512, respectively, where we derive the corresponding reduced-round integral distinguishers under the considered settings. In Section 6, we show how these distinguishers can be extended to key-recovery attacks on reduced-round variants of Kalyna. Section 7 discusses the significance and limitations of the obtained results. Finally, Section 8 concludes the paper. For completeness, the detailed propagation tables and auxiliary technical material are provided in the appendices.

2. Preliminaries

In this section, we briefly introduce the notation and basic concepts used throughout the paper. We first recall the integral properties employed in our analysis and then describe the round structure of Kalyna for its standard block sizes 128, 256, and 512 bits. Let n be a positive integer. We denote by $\mathbb{F}_2$ the prime field of characteristic 2, and by $\mathbb{F}_{2^n}$ an extension field of degree n over $\mathbb{F}_2$. The set of all nonzero elements of $\mathbb{F}_{2^n}$ is denoted by $\mathbb{F}_{2^n}^*$, while $\mathbb{F}_2^n$ denotes the vector space of all n-tuples over $\mathbb{F}_2$. An element of $\mathbb{F}_2^n$ is written as $x = (x_1, x_2, \ldots, x_n)$,

Table 1: Comparison with previous public results on reduced-round Kalyna.

Variant	Attack type	Rounds attacked	Data (CP/CC)	Time (Enc.)	Memory	Reference
Kalyna-128/128	Integral (with pre-whitening)	4 (distinguisher)	2^{96} CPs	–	–	[13]
	Key-dependent integral	4 (distinguisher)	2^{62} CPs	–	–	[13]
	Integral (without pre-whitening)	4 (distinguisher)	2^{64} CPs	–	–	[13]
	Integral	$3^{\ddagger}$ (key-recovery)	2^{9} CPs	2^{74}	negligible	This work, Alg. 2
	Weak-key integral	$4^{\S}$ (key-recovery)	2^{9} CPs	2^{74}	negligible	This work, Alg. 1
	Integral(without pre-whitening)	$5^{\dagger}$ (key-recovery)	2^{9} CPs	2^{74}	negligible	This work, Alg. 2
Kalyna-128/256	Meet-in-the-middle	7 (key-recovery)	2^{89} CPs	$2^{230.2}$	$2^{202.64}$	[11]
	Multiset / MITM	9 (key-recovery)	2^{105} CPs	$2^{245.83}$	$2^{226.86}$	[11]
	Meet-in-the-middle	9 (key-recovery)	2^{127} CPs	$2^{208.38}$	2^{204}	[10]
	Integral	$4^{\ddagger}$ (key-recovery)	2^{9} CPs	2^{149}	negligible	This work, Alg. 4
	Weak-key integral	$5^{\S}$ (key-recovery)	2^{9} CPs	2^{137}	negligible	This work, Alg. 3
	Integral(without pre-whitening)	$6^{\dagger}$ (key-recovery)	2^{9} CPs	2^{149}	negligible	This work, Alg. 4
Kalyna-256/256	Integral	$3^{\ddagger}$ (key-recovery)	2^{9} CPs	2^{75}	negligible	This work, Alg. 2
	Weak-key integral	$4^{\S}$ (key-recovery)	2^{17} CPs	2^{83}	negligible	This work, Alg. 1
	Weak-key integral	$5^{\S}$ (key-recovery)	2^{17} CPs	2^{83}	negligible	This work, Alg. 2
Kalyna-256/512	Meet-in-the-middle	7 (key-recovery)	2^{233} CPs	$2^{502.2}$	2^{170}	[11]
	Multiset / MITM	9 (key-recovery)	2^{217} CPs	$2^{477.83}$	$2^{443.45}$	[12]
	Meet-in-the-middle	11 (key-recovery)	2^{233} CPs	2^{425}	$2^{403.5}$	[10]
	Weak-key integral	$4^{\S}$ (key-recovery)	2^{9} CPs	2^{278}	negligible	This work, Alg. 4
	Weak-key integral	$5^{\S}$ (key-recovery)	2^{17} CPs	2^{273}	negligible	This work, Alg. 3
	Weak-key integral	$6^{\S}$ (key-recovery)	2^{17} CPs	2^{286}	negligible	This work, Alg. 4
Kalyna-512/512	Integral	$4^{\ddagger}$ (key-recovery)	2^{9} CPs	2^{76}	negligible	This work, Alg. 1
	Weak-key integral	$5^{\S}$ (key-recovery)	2^{9} CPs	2^{76}	negligible	This work, Alg. 2
	Integral(without pre-whitening)	$6^{\dagger}$ (key-recovery)	2^{9} CPs	2^{76}	negligible	This work, Alg. 1

† Without pre-whitening operation.
§ With pre-whitening operation with weak key space.
‡ A key-independent distinguisher.

where $x_i \in \mathbb{F}_2$, and throughout the paper, after fixing a basis, we identify $\mathbb{F}_{2^n}$ with $\mathbb{F}_2^n$ via the corresponding standard vector space isomorphism over $\mathbb{F}_2$. These preliminaries will be used in the subsequent sections to derive reduced-round integral distinguishers and their corresponding key-recovery extensions.

Throughout this paper, we use the following notation. We also recall several basic definitions that will be used in the subsequent analysis.

0	8
1	9
2	10
3	11
4	12
5	13
6	14
7	15

(a) Kalyna-128

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

(b) Kalyna-256

0	8	16	24	32	40	48	56
1	9	17	25	33	41	49	57
2	10	18	26	34	42	50	58
3	11	19	27	35	43	51	59
4	12	20	28	36	44	52	60
5	13	21	29	37	45	53	61
6	14	22	30	38	46	54	62
7	15	23	31	39	47	55	63

(c) Kalyna-512

Figure 1: Column-wise byte numbering in Kalyna states.

$\mathcal{M}_b^{(r)}$	:	Chosen input multiset used to derive an r-round integral distinguisher, where $b \in \{128, 256, 512\}$ denotes the block size.
$\mathcal{C}_b^{(r)}$	:	The set of ciphertexts obtained after applying r rounds to all elements of the input multiset.
K^{pw}	:	Pre-whitening key. In the Kalyna-128 setting, $K^{\mathrm{pw}} = (k_0, k_1, \ldots, k_{15})$.
$\mathcal{K}_{\mathrm{wk}}$	:	Weak-key space associated with the considered distinguisher. For Kalyna-128, $\mathcal{K}_{\mathrm{wk}} = \left\{ K^{\mathrm{pw}} \in (\mathbb{F}_2^8)^{16} \mid k_i = 0, k_j = 0,\ 0 \le i, j \le 15 \right\}$.
$\lvert\mathcal{K}_{\mathrm{wk}}\rvert$	:	Cardinality of the weak-key space $\mathcal{K}_{\mathrm{wk}}$.
c_i	:	Constant byte value at byte position i in a chosen input multiset.
$\oplus$	:	Bitwise XOR.
$\boxplus$	:	Addition modulo 2^{64}.
$\boxminus$	:	Subtraction modulo 2^{64}.
$\bigoplus_{x \in \mathcal{M}_b^{(r)}} X_s[i] = 0$	:	i-th byte of the state X_s is balanced over the multiset $\mathcal{M}_b^{(r)}$.
$K^{(r)}$	:	Round key of round r.
$K_j^{(r)}$	:	jth 64-bit column of the round key $K^{(r)}$.
C_j	:	jth ciphertext column.
$\widehat{K}_j^{(r)}$	:	A guess for the jth 64-bit column of the round key $K^{(r)}$.
U_j	:	Intermediate value defined by $U_j = C_j \boxminus \widehat{K}_j^{(r)}$.
V_j	:	Intermediate value obtained after applying the inverse MixColumns, i.e., $V_j = \mathsf{MC}^{-1}(U_j)$.
$\mathsf{SB}, \mathsf{SR}, \mathsf{MC}$	:	SubBytes, ShiftRows, and MixColumns operations, respectively.
$\mathsf{SB}^{-1}, \mathsf{SR}^{-1}, \mathsf{MC}^{-1}$	:	Inverse SubBytes, inverse ShiftRows, and inverse MixColumns operations, respectively.

We use the standard terminology of integral cryptanalysis. For a given byte position in a chosen multiset, we write:

- **All** (A), if every possible byte value occurs equally often;
- **Balanced** (B), if the XOR sum of all values taken by that byte over the multiset is zero;
- **Constant** (C), if the byte takes a fixed value throughout the multiset;
- **Unknown** (U), if none of the above properties is guaranteed.

We use the standard integral notation $A/C/B/U$, where A, C, B and U mean All, Constant, Balanced, and Unknown, respectively.

2.1. Description of Kalyna

Kalyna is a substitution–permutation network block cipher standardized in Ukraine [14]. It supports block sizes of 128, 256, and 512 bits, while the key size is either equal to the block size

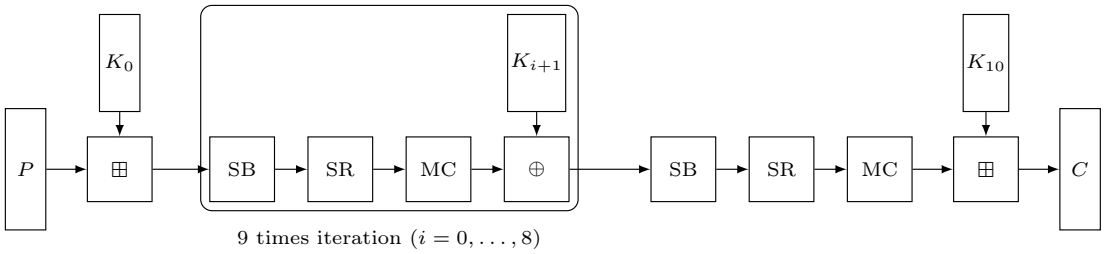


Figure 2: One full encryption of Kalyna-128/128.

or twice the block size. Accordingly, the five standard variants are Kalyna-128/128, Kalyna-128/256, Kalyna-256/256, Kalyna-256/512, and Kalyna-512/512. The corresponding numbers of rounds are 10, 14, 14, 18, and 18, respectively [14]. The internal state is represented as a byte matrix of size 8×2, 8×4, and 8×8 for the 128-bit, 256-bit, and 512-bit versions, respectively, where the bytes are numbered column-wise. Figure 1 illustrates this byte numbering for the three state sizes. The same description is commonly used in later cryptanalytic works on Kalyna as well [12].

Before the first round, Kalyna applies a pre-whitening operation, and after the last round it applies a post-whitening operation. In both cases, the key addition is performed column-wise by addition modulo 2^{64} [14]. A full encryption structure of Kalyna is shown in Figure 2, which highlights the whitening layers and the sequence of round transformations. Each internal round consists of the following four basic operations:

$$\mathsf{SB} \rightarrow \mathsf{SR} \rightarrow \mathsf{MC} \rightarrow \mathsf{ARK}.$$

In the notation of the original specification, these correspond to the nonlinear substitution layer, the row permutation layer, the linear diffusion layer, and the round-key addition, respectively [14].

- SubBytes (SB): This is the nonlinear substitution layer. Kalyna uses four different 8-bit S-boxes, denoted by S_0, S_1, S_2, and S_3. A byte in row j is substituted using the S-box $S_{j \bmod 4}$ [14].
- ShiftRows (SR): This operation cyclically shifts the bytes within each row. The amount of shift depends on the row index and the block size. More precisely, if the block size is $l \in \{128, 256, 512\}$, then the shift amount for row i is given by

  $$\delta_i = \left\lfloor \frac{i \cdot l}{512} \right\rfloor.$$

  The inverse ShiftRows operation performs the corresponding cyclic shift in the opposite direction [14].
- MixColumn (MC): This is the linear diffusion layer. Each column of the state is multiplied by an 8×8 MDS matrix over $\mathbb{F}_{2^8}$, where the field is defined by the irreducible polynomial $x^8 + x^4 + x^3 + x^2 + 1$. This layer provides strong diffusion across the state bytes [14].
- AddRoundKey (ARK): In this step, the state is XORed with the round subkey. The round subkeys have the same size as the internal state [14].

Thus, Kalyna combines byte-wise nonlinearity through the substitution layer and strong diffusion through the permutation and MDS-based linear layer, together with modular key addition in the whitening stage. These design features make Kalyna structurally similar to AES in some respects, while the larger MDS layer, multiple S-boxes, and modulo 2^{64} whitening

operations give it a distinct cryptographic structure [14, 12]. For the convenience of the reader, we provide in Appendix Appendix A the four forward S-boxes, the corresponding inverse S-boxes, and the MDS matrix defining the linear diffusion layer together with its inverse.

Before the first round, Kalyna applies a pre-whitening operation, and after the last round it applies a post-whitening operation. In both cases, the key addition is performed independently on each 64-bit column by addition modulo 2^{64}. The internal rounds, however, use XOR with the corresponding round subkeys.For a 64-bit column written as $(x_0, x_1, \ldots, x_7)$, Kalyna uses the little-endian convention in the modular addition/subtraction layers. Therefore, x_0 is the least significant byte (LSB) and x_7 is the most significant byte (MSB) of the column [14]. The following observation will be useful in our later analysis.

Lemma 1. *In Kalyna, if a 64-bit column has one active byte at its most significant byte position and all remaining seven bytes are constant, then the pre-whitening addition modulo 2^{64} preserves the corresponding A/C structure.*

Proof. The pre-whitening operation is performed independently on each 64-bit column by addition modulo 2^{64} using little-endian byte order. Hence, the most significant byte is affected only by a fixed carry coming from the lower seven bytes. If these lower seven bytes are constant over the whole multiset, then the carry into the most significant byte is also fixed. Consequently, the most significant byte is transformed by addition with a fixed constant modulo 2^8, which is a bijection, and therefore its *All* property is preserved. The other seven bytes remain constant as well. □

Lemma 2. *Let $u = (u_0, u_1, \ldots, u_7)^T$ be an input column of Kalyna, and let $M = (m_{j,i})$ be the MixColumns matrix. Assume that the active components of u are indexed by $I = \{i_1, i_2, \ldots, i_t\}$, $t \geq 1$, and that these active components are given by t-independent variables $x_1, x_2, \ldots, x_t \in \mathbb{F}_{2^8}$, while all remaining components are constant. Then every component of the output column $v = Mu$ has property A.*

Proof. By assumption, the input column may be written as

$$u_i = \begin{cases} x_r, & i = i_r \text{ for some } r \in \{1, 2, \ldots, t\}, \\ c_i, & i \notin I, \end{cases}$$

where $x_1, x_2, \ldots, x_t$ are independent and the c_i's are constants. For each $0 \leq j \leq 7$, the j-th output byte is

$$v_j = \bigoplus_{i=0}^{7} m_{j,i} u_i = \left(\bigoplus_{i \notin I} m_{j,i} c_i \right) \oplus \bigoplus_{r=1}^{t} m_{j,i_r} x_r.$$

Let $d_j = \bigoplus_{i \notin I} m_{j,i} c_i$. Then we have $v_j = d_j \oplus \bigoplus_{r=1}^{t} m_{j,i_r} x_r$. Since every entry of the Kalyna MixColumns matrix is nonzero, for each active position i_r and each output position j we have $m_{j,i_r} \neq 0$. Hence

$$v_j = d_j \oplus m_{j,i_1} x_1 \oplus \cdots \oplus m_{j,i_t} x_t$$

is a nonconstant affine function of independent variables $x_1, x_2, \ldots, x_t$. Fix any value $b \in \mathbb{F}_{2^8}$. The equation

$$d_j \oplus m_{j,i_1} x_1 \oplus \cdots \oplus m_{j,i_t} x_t = b$$

is equivalent to one nontrivial affine equation in the t-independent variables $x_1, \ldots, x_t$. Therefore it has exactly $2^{8(t-1)}$ solutions in $\mathbb{F}_{2^8}^t$. Thus each value of $\mathbb{F}_{2^8}$ occurs equally often as an output value of v_j. Hence v_j has property A. Since this holds for every j, all eight output bytes have property A. □

The conclusion of Lemma 2 need not hold if the active input bytes are not independent. In particular, if the same active variable appears in several positions, cancellation may occur, and some output bytes may fail to have property A.

3. Integral analysis of Kalyna-128

In this section, we study the propagation of integral properties through the round function of Kalyna. We first focus on the Kalyna-128 variant and derive reduced-round integral distinguishers from carefully chosen input multisets. More precisely, we obtain a 2-round integral distinguisher, a 3-round weak-key integral distinguisher under a suitable assumption on the pre-whitening layer, and a 4-round integral distinguisher in the setting where the pre-whitening part is omitted.

3.1. 2-round integral distinguisher

In this subsection, we present a 2-round integral distinguisher for Kalyna-128. Recall that the internal state of Kalyna-128 is represented as an 8×2 byte matrix, where the bytes are numbered column-wise. We begin with the following multiset of size 2^8, in which the 7th byte of the first column is active and takes all 256 possible values, while the remaining 15 bytes are fixed to constants:

$$\mathcal{M}_{128}^{(2)} = \begin{pmatrix} C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ A & C \end{pmatrix}.$$

Since the active byte is placed at the most significant byte position of a 64-bit column and the remaining seven bytes in that column are constant, it follows from Lemma 1 that the pre-whitening operation preserves the above A/C structure. Hence, the multiset keeps the same integral structure after the pre-whitening step.

The SubBytes layer preserves the A/C structure, since each S-box is a bijection on $\mathbb{F}_2^8$. Moreover, for Kalyna-128, the ShiftRows operation shifts rows $0, 1, 2, 3$ by 0 positions and rows $4, 5, 6, 7$ by 1 position. Therefore, the active byte located at row 7 of the first column is moved to row 7 of the second column. Thus, just before the MixColumns layer of the first round, the multiset becomes

$$\begin{pmatrix} C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & A \end{pmatrix}.$$

Now the second column contains exactly one active byte and seven constant bytes. Hence, by Lemma 2, the MixColumns operation maps this column to a column in which all eight bytes have property A. Therefore, after the first round, the multiset takes the form

$$\begin{pmatrix} C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \end{pmatrix}.$$

Starting from this intermediate state, after the second round of Kalyna-128, all 16 bytes of the output state satisfy the *Balanced* property, and after two rounds we obtain

$$\begin{pmatrix} B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \end{pmatrix}.$$

Therefore, Kalyna-128 admits a 2-round integral distinguisher with data complexity 2^8 when the active byte is chosen at the 7th byte of the first column. In this distinguisher, all 16 output bytes are balanced. For completeness, the detailed propagation of the multiset $\mathcal{M}_{128}^{(2)}$ through the pre-whitening layer and the subsequent two rounds of Kalyna-128 is provided in Appendix Appendix B.

3.2. 3-round weak-key integral distinguisher

We now extend the 2-round integral distinguisher of the previous subsection to 3 rounds under a weak-key assumption. Recall that the internal state of Kalyna-128 is represented as an 8×2 byte matrix, with bytes numbered column-wise, and that the multiset

$$\mathcal{M}_{128}^{(2)} = \begin{pmatrix} C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ A & C \end{pmatrix}$$

of size 2^8 yields a 2-round integral distinguisher.

To extend this property by one round backward, we apply the inverse round transformation

$$\mathrm{SB}^{-1} \circ \mathrm{SR}^{-1} \circ \mathrm{MC}^{-1}.$$

Since XOR with a fixed round key does not affect the $A/C/B/U$ structure, the inverse round-key addition can be omitted from the structural discussion. The inverse linear layer redistributes the activity within each column, the inverse ShiftRows only permutes byte positions, and the inverse substitution layer is bytewise bijective. Hence, we obtain the backward-derived multiset

$$\mathcal{M}_{128}^{(3)} = \begin{pmatrix} A & C \\ A & C \\ A & C \\ A & C \\ C & A \\ C & A \\ C & A \\ C & A \end{pmatrix}.$$

Equivalently, bytes $0, 1, 2, 3, 12, 13, 14, 15$ satisfy the *All* property, while bytes $4, 5, 6, 7, 8, 9, 10, 11$ are constant.

We use $\mathcal{M}_{128}^{(3)}$ as the input structure for the 3-round analysis. The main obstacle is the pre-whitening operation, which performs addition modulo 2^{64} on each 64-bit column and may destroy the bytewise integral structure through carry propagation. Write the pre-whitening key as $K^{\mathrm{pw}} = (k_0, k_1, \ldots, k_{15})$, where $(k_0, \ldots, k_7)$ and $(k_8, \ldots, k_{15})$ correspond to the two state columns. We consider the weak-key space $\mathcal{K}_{\mathrm{wk}} = \Big\{ K^{\mathrm{pw}} \in (\mathbb{F}_2^8)^{16} \mid k_0 = k_1 = k_2 = k_3 = k_{12} = k_{13} = k_{14} = k_{15} = 0 \Big\}$. Since 8 bytes are fixed and the remaining 8 bytes are arbitrary, we have $|\mathcal{K}_{\mathrm{wk}}| = 2^{64}$, so $\mathcal{K}_{\mathrm{wk}}$ is a weak-key space of size 2^{64}.

We further choose the constants in the plaintext structure such that $c_8 = c_9 = c_{10} = c_{11} = 0$. Then the effect of pre-whitening can be analyzed column by column. In the first column, bytes $0, 1, 2, 3$ are active and bytes $4, 5, 6, 7$ are constant. Since $k_0 = k_1 = k_2 = k_3 = 0$, the lower 32-bit part of the column is unchanged, so no carry propagates from byte 3 into byte 4, and bytes $4, 5, 6, 7$ remain constant after adding the fixed key bytes k_4, k_5, k_6, k_7.

In the second column, bytes $8, 9, 10, 11$ are constant and bytes $12, 13, 14, 15$ are active. Because $c_8 = c_9 = c_{10} = c_{11} = 0$ and $k_{12} = k_{13} = k_{14} = k_{15} = 0$, the lower 32-bit part becomes the fixed constant tuple $(k_8, k_9, k_{10}, k_{11})$, while no carry is generated from byte 11 into byte 12. Moreover, the active bytes $12, 13, 14, 15$ are not modified by the key addition itself. Therefore, the A/C pattern of the second column is also preserved.

Hence, for every key in $\mathcal{K}_{\mathrm{wk}}$, the pre-whitening layer maps $\mathcal{M}_{128}^{(3)}$ to a multiset with the same integral structure. By construction, after one forward round this multiset evolves into the reference multiset $\mathcal{M}_{128}^{(2)}$. From the previous subsection, $\mathcal{M}_{128}^{(2)}$ yields a 2-round integral distinguisher for which all 16 output bytes are balanced. Therefore, under the weak-key assumption $K^{\mathrm{pw}} \in \mathcal{K}_{\mathrm{wk}}$, $\mathcal{M}_{128}^{(3)}$ gives a 3-round integral distinguisher for Kalyna-128, and after 3 rounds we obtain

$$\begin{pmatrix} B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \end{pmatrix}.$$

Thus, Kalyna-128 admits a 3-round weak-key integral distinguisher with data complexity 2^8 and weak-key space 2^{64}, for which all 16 output bytes are balanced. The detailed propagation of $\mathcal{M}_{128}^{(3)}$ through the pre-whitening layer and the three rounds of Kalyna-128 is given in Appendix Appendix C.

3.3. 4-round integral distinguisher without pre-whitening

We now extend the previous construction by one more round and obtain a 4-round integral distinguisher for Kalyna-128 when the pre-whitening operation is omitted. Recall that, in the 3-round case, we derived the multiset

$$\mathcal{M}_{128}^{(3)} = \begin{pmatrix} A & C \\ A & C \\ A & C \\ A & C \\ C & A \\ C & A \\ C & A \\ C & A \end{pmatrix}.$$

To extend the distinguisher by one round, we propagate this multiset backward through one inverse round transformation

$$\mathrm{SB}^{-1} \circ \mathrm{SR}^{-1} \circ \mathrm{MC}^{-1}.$$

As before, since XOR with a fixed round key does not change the integral type of a byte, the inverse round-key addition does not affect the structural discussion. Applying the above inverse transformation once more to $\mathcal{M}_{128}^{(3)}$ gives a new multiset, denoted by $\mathcal{M}_{128}^{(4)}$. The multiset $\mathcal{M}_{128}^{(4)}$ still has size 2^8, since the inverse round transformation is bijective. However, unlike $\mathcal{M}_{128}^{(3)}$, it does not admit a simple description in terms of the bytewise symbols A, B, C, and U. Thus, $\mathcal{M}_{128}^{(4)}$ should be regarded as an exact multiset obtained by inverse propagation. We now use $\mathcal{M}_{128}^{(4)}$ as the input of a 4-round version of Kalyna-128 without pre-whitening. This assumption is necessary because, in contrast to the 3-round weak-key case, we do not impose any restriction on the whitening key to preserve the required structure under addition modulo 2^{64}.

By construction, after one forward round, $\mathcal{M}_{128}^{(4)}$ evolves into $\mathcal{M}_{128}^{(3)}$. Then, by the same propagation argument as in the 3-round case, after the next three rounds every byte of the output state is balanced. Consequently, after 4 rounds we obtain

$$\begin{pmatrix} B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \end{pmatrix}.$$

Therefore, when the pre-whitening operation is omitted, Kalyna-128 admits a 4-round integral distinguisher with data complexity 2^8, for which all 16 output bytes are balanced. The detailed propagation of $\mathcal{M}_{128}^{(4)}$ through the four rounds of Kalyna-128 is given in Appendix Appendix D.

4. Integral analysis of Kalyna-256

In this section, we extend the integral analysis to the Kalyna-256 variant. Recall that the internal state of Kalyna-256 is represented as an 8×4 byte matrix. Using suitably chosen input multisets, we study the propagation of integral properties through the round function and derive reduced-round integral distinguishers for Kalyna-256.

4.1. 2-round integral distinguisher

In this subsection, we present a 2-round integral distinguisher for Kalyna-256. The internal state of Kalyna-256 is represented as an 8×4 byte matrix with bytes numbered column-wise. In particular, the byte positions $7, 15, 23$, and 31 are the most significant bytes of the four 64-bit columns, respectively. The pre-whitening operation is applied independently to each 64-bit column by addition modulo 2^{64} in little-endian byte order. We consider multisets of size 2^8 in which one of the byte positions $7, 15, 23$, or 31 is active and takes all 256 possible values, while the remaining 31 bytes are constant. Since these four cases are equivalent by column symmetry, it suffices to consider the representative multiset

$$\mathcal{M}_{256}^{(2)} = \begin{pmatrix} C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ A & C & C & C \end{pmatrix}.$$

Because the active byte is placed at the most significant byte position of a 64-bit column and the other seven bytes in that column are constant, Lemma 1 implies that the pre-whitening operation preserves this A/C structure.

The ShiftRows operation of Kalyna-256 uses the row-shift pattern $(0, 0, 1, 1, 2, 2, 3, 3)$. Thus, in the first round, the active byte in row 7 is moved to another column by ShiftRows, and the subsequent MixColumn layer diffuses it to all eight bytes of that column. In the second round, the substitution layer preserves the integral structure, while ShiftRows redistributes these eight active bytes so that each column contains two active positions. After the second MixColumn layer, every output byte becomes balanced. Since the round-key addition inside a round is XOR with a fixed subkey, it also preserves the *Balanced* property. Hence, after two rounds we obtain

$$\begin{pmatrix} B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \end{pmatrix}.$$

Therefore, Kalyna-256 admits a 2-round integral distinguisher with data complexity 2^8 when the active byte is chosen at any one of the positions $7, 15, 23$, or 31. In each case, all 32 output bytes are balanced. The detailed propagation of $\mathcal{M}_{256}^{(2)}$ through the pre-whitening layer and the following two rounds of Kalyna-256 is given in Appendix Appendix E.

4.2. 3-round weak-key integral distinguisher

In this subsection, we present a 3-round weak-key integral distinguisher for Kalyna-256. Unlike the Kalyna-128 case, here we do not construct the 3-round distinguisher by propagating the 2-round input structure one round backward. Such a backward extension would require a more restrictive input pattern and, consequently, stronger conditions on the pre-whitening key in order to preserve the desired integral structure. To keep the weak-key assumption as large as possible, we instead adopt a forward construction with two independent active bytes. These two bytes are chosen so that, after the first ShiftRows layer, they are moved into the same column. For this reason, the input multiset considered in this subsection has size 2^{16}. We therefore consider a multiset of size 2^{16} in which two bytes are active and take all possible values independently, while the remaining 30 bytes are fixed to constants. One active byte is chosen at a most significant byte position of a 64-bit column, namely at one of the positions $7, 15, 23$, or 31. The second active byte is chosen so that, after the ShiftRows operation of the first round, both active bytes are moved into the same column. For example, when the first active byte is placed at position 7, the second active byte may be chosen from $\{24, 25, 18, 19, 12, 13, 6\}$. The other cases are equivalent by symmetry.

For concreteness, we consider the representative case in which the active bytes are placed at positions 7 and 24. The input multiset is

$$\mathcal{M}_{256}^{(3)} = \begin{pmatrix} C & C & C & A \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ A & C & C & C \end{pmatrix}.$$

Let $K^{\mathrm{pw}} = (k_0, k_1, \ldots, k_{31})$ be the pre-whitening key. Since the byte at position 7 is the most significant byte of the first 64-bit column, Lemma 1 implies that the corresponding A/C structure is preserved under the pre-whitening addition modulo 2^{64}, independently of the whitening key. For the second active byte at position 24, we impose the weak-key condition $k_{24} = 0$. Equivalently, we consider the weak-key space

$$\mathcal{K}_{\mathrm{wk}} = \left\{ K^{\mathrm{pw}} \in (\mathbb{F}_2^8)^{32} \;\middle|\; k_{24} = 0 \right\},$$

which has size $|\mathcal{K}_{\mathrm{wk}}| = 2^{248}$. Under this condition, the active byte at position 24 is unchanged by the pre-whitening addition. Moreover, since the addition at byte 24 is of the form $x \boxplus 0 = x$, no carry is generated from byte 24 to byte 25, so the bytes $25, \ldots, 31$ remain constant. Hence, for every key in $\mathcal{K}_{\mathrm{wk}}$, the multiset $\mathcal{M}_{256}^{(3)}$ is mapped by pre-whitening to a multiset with the same active/constant structure.

By construction, after the ShiftRows operation of the first round, the two active bytes lie in the same column. Since these active bytes are independent, Lemma 2 implies that the first MixColumn transformation maps this column to a column whose eight bytes all have property A. In the second round, the substitution layer preserves this structure, while ShiftRows redistributes these eight active positions among the four columns. Hence, before the second MixColumn transformation, each of the four columns contains two active bytes depending on the same two independent active variables. Applying Lemma 2 again, we conclude that after the second MixColumn transformation every byte of the state has property A. Since the *All* property implies the *Balanced* property, every output byte is balanced after three rounds. Hence, after three rounds we obtain

$$\begin{pmatrix} B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \end{pmatrix}.$$

Therefore, Kalyna-256 admits a 3-round weak-key integral distinguisher with data complexity 2^{16} and weak-key space 2^{248} in the representative case above, for which all 32 output bytes are balanced. The other admissible choices of the second active byte can be treated analogously. The detailed propagation of the representative multiset through the pre-whitening layer and the subsequent three rounds of Kalyna-256 is given in Appendix Appendix F.

4.3. 4-round weak-key integral distinguisher

In this subsection, we extend the previous construction by one more round and obtain a 4-round weak-key integral distinguisher for Kalyna-256. Recall that, in the previous subsection, we considered the input multiset

$$\mathcal{M}_{256}^{(3)} = \begin{pmatrix} C & C & C & A \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ A & C & C & C \end{pmatrix},$$

which yields a 3-round weak-key integral distinguisher for Kalyna-256. To extend the distinguisher by one round, we propagate the above multiset backward through one inverse round transformation

$$\mathrm{SB}^{-1} \circ \mathrm{SR}^{-1} \circ \mathrm{MC}^{-1}.$$

As before, since XOR with a fixed round subkey does not change the integral type of a byte, the inverse round-key addition does not affect the structural discussion. Applying this inverse transformation to $\mathcal{M}_{256}^{(3)}$, we obtain

$$\mathcal{M}_{256}^{(4)} = \begin{pmatrix} A & C & C & A \\ A & C & C & A \\ C & C & A & A \\ C & C & A & A \\ C & A & A & C \\ C & A & A & C \\ A & A & C & C \\ A & A & C & C \end{pmatrix}.$$

Since the inverse round transformation is bijective, $\mathcal{M}_{256}^{(4)}$ also has size 2^{16}.

We now use $\mathcal{M}_{256}^{(4)}$ as the input structure for the 4-round analysis including the pre-whitening layer. Since the pre-whitening operation of Kalyna-256 is defined by addition modulo 2^{64} on each 64-bit column, special care is needed to preserve the bytewise A/C structure. To this end, we choose the fixed constants in the plaintext structure such that $c_2 = c_3 = c_4 = c_5 = c_8 = c_9 = c_{10} = c_{11} = c_{16} = c_{17} = 0$. Let $K^{\mathrm{pw}} = (k_0, k_1, \ldots, k_{31})$ be the pre-whitening key. We consider the weak-key space

$$\mathcal{K}_{\mathrm{wk}} = \left\{ K^{\mathrm{pw}} \in (\mathbb{F}_2^8)^{32} \;\middle|\; \begin{array}{l} k_0 = k_1 = k_6 = k_7 = 0,\ k_{12} = k_{13} = k_{14} = k_{15} = 0, \\ k_{18} = k_{19} = k_{20} = k_{21} = 0,\ k_{24} = k_{25} = k_{26} = k_{27} = 0 \end{array} \right\},$$

for which $|\mathcal{K}_{\mathrm{wk}}| = 2^{128}$. Under these conditions, the pre-whitening layer preserves the A/C pattern of $\mathcal{M}_{256}^{(4)}$. Indeed, the imposed zero conditions on the selected constant bytes prevent unwanted carries from propagating from the lower constant parts of the relevant 64-bit columns into the active bytes, while the zero conditions on the specified key bytes prevent the active parts from generating carries that would destroy the required structure. Hence, for every key in $\mathcal{K}_{\mathrm{wk}}$, the multiset $\mathcal{M}_{256}^{(4)}$ is mapped by pre-whitening to a multiset with the same bytewise A/C structure.

By construction, after one forward round, $\mathcal{M}_{256}^{(4)}$ evolves into $\mathcal{M}_{256}^{(3)}$. Moreover, the weak-key space $\mathcal{K}_{\mathrm{wk}}$ is compatible with the weak-key condition used in the previous subsection. Therefore, the same 3-round propagation argument applies to $\mathcal{M}_{256}^{(3)}$, and all 32 output bytes are balanced after the next three rounds. Consequently, after four rounds we obtain

$$\begin{pmatrix} B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \end{pmatrix},$$

that is, every byte of the output state is balanced. Therefore, Kalyna-256 admits a 4-round weak-key integral distinguisher with data complexity 2^{16} and weak-key space 2^{128}, for which all 32 output bytes are balanced. The detailed propagation of $\mathcal{M}_{256}^{(4)}$ through the pre-whitening layer and the subsequent four rounds of Kalyna-256 is given in Appendix Appendix G.

5. Integral analysis of Kalyna-512

In this section, we extend the integral analysis to the Kalyna-512 variant. Recall that the internal state of Kalyna-512 is represented as an 8×8 byte matrix. Using suitably chosen input multisets, we study the propagation of integral properties through the round function and derive reduced-round integral distinguishers for Kalyna-512.

5.1. 3-round integral distinguisher

In this subsection, we present a 3-round integral distinguisher for Kalyna-512. The internal state of Kalyna-512 is represented as an 8×8 byte matrix with bytes numbered column-wise. In particular, the byte positions $7, 15, 23, 31, 39, 47, 55, 63$ are the most significant bytes of the

eight 64-bit columns. We consider a multiset of size 2^8 in which one of these byte positions is active and takes all 256 possible values, while the remaining 63 bytes are fixed to constants. Since these choices are equivalent by column symmetry, it suffices to consider the representative case in which the active byte is placed at position 7. Thus, we start from

$$\mathcal{M}_{512}^{(3)} = \begin{pmatrix} C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ A & C & C & C & C & C & C & C \end{pmatrix}.$$

Since the active byte is placed at the most significant byte position of a 64-bit column and the remaining seven bytes in that column are constant, Lemma 1 implies that the pre-whitening operation preserves this A/C structure.

We now trace the propagation of this multiset through three rounds of Kalyna-512. In the first round, the substitution layer SB preserves the active byte, since the S-boxes are bijective. The ShiftRows operation of Kalyna-512 uses the row-shift pattern $(0, 1, 2, 3, 4, 5, 6, 7)$. Hence, the active byte in row 7 is moved to another column by SR, and the subsequent linear layer MC diffuses this activity to all eight bytes of that column by Lemma 2. Therefore, after the first round, one entire column satisfies the *All* property.

In the second round, SB again preserves the integral structure, while SR redistributes these eight active bytes into eight distinct columns, since the row shifts $0, 1, \ldots, 7$ are all different. Thus, after SR, each column contains exactly one active byte. Applying MC to each column and using Lemma 2 again, this activity diffuses to all eight bytes of each column. Hence, after the second round, all 64 bytes of the state satisfy the *all* property.

In the third round, both SB and SR preserve the *All* property of every byte. Since every *All* byte is also balanced, each input byte to the final linear layer MC is balanced. Because MC is linear, every output byte is a linear combination of balanced quantities and is therefore balanced. The final round-key addition, being an XOR with a fixed subkey, also preserves the *Balanced* property. Therefore, after three rounds we obtain

$$\begin{pmatrix} B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \end{pmatrix}.$$

Therefore, Kalyna-512 admits a 3-round integral distinguisher with data complexity 2^8 when the active byte is chosen at any one of the positions $7, 15, 23, 31, 39, 47, 55$, or 63. In each case, all 64 output bytes are balanced. The detailed propagation of the chosen multiset through the pre-whitening layer and the subsequent three rounds of Kalyna-512 is given in Appendix Appendix H.

5.2. 4-round weak-key integral distinguisher

In this subsection, we extend the previous construction by one more round and obtain a 4-round weak-key integral distinguisher for Kalyna-512. Recall that, in the previous subsection, we considered the multiset

$$\mathcal{M}_{512}^{(3)} = \begin{pmatrix} C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ A & C & C & C & C & C & C & C \end{pmatrix},$$

which yields a 3-round integral distinguisher for Kalyna-512. To extend the distinguisher by one round, we propagate the above multiset backward through one inverse round transformation

$$\mathrm{SB}^{-1} \circ \mathrm{SR}^{-1} \circ \mathrm{MC}^{-1}.$$

As before, since XOR with a fixed round subkey does not change the integral type of a byte, the inverse round-key addition does not affect the structural discussion. Applying this transformation to $\mathcal{M}_{512}^{(3)}$, we obtain

$$\mathcal{M}_{512}^{(4)} = \begin{pmatrix} A & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & A & C \\ C & C & C & C & C & A & C & C \\ C & C & C & C & A & C & C & C \\ C & C & C & A & C & C & C & C \\ C & C & A & C & C & C & C & C \\ C & A & C & C & C & C & C & C \end{pmatrix}.$$

Since the inverse round transformation is bijective, $\mathcal{M}_{512}^{(4)}$ also has size 2^8.

We now use $\mathcal{M}_{512}^{(4)}$ as the input structure for the 4-round analysis including the pre-whitening layer. In this multiset, the active byte positions are 0, 15, 22, 29, 36, 43, 50, 57. To preserve the required bytewise A/C structure under pre-whitening, we must prevent unwanted carries from the lower constant parts of the corresponding 64-bit columns into these active bytes. For this purpose, we choose the fixed constants in the plaintext structure such that

$$c_{16} = c_{17} = c_{18} = c_{19} = c_{20} = c_{21} = 0,$$
$$c_{24} = c_{25} = c_{26} = c_{27} = c_{28} = 0,$$
$$c_{32} = c_{33} = c_{34} = c_{35} = 0,$$
$$c_{40} = c_{41} = c_{42} = 0, \qquad c_{48} = c_{49} = 0, \qquad c_{56} = 0.$$

Under these choices, no carry propagates from the lower constant parts of the corresponding columns into the active bytes. Let $K^{\mathrm{pw}} = (k_0, k_1, \ldots, k_{63})$ be the pre-whitening key. We consider the weak-key space $\mathcal{K}_{\mathrm{wk}} = \left\{ K^{\mathrm{pw}} \in (\mathbb{F}_2^8)^{64} \;\middle|\; k_0 = k_{15} = k_{22} = k_{29} = k_{36} = k_{43} = k_{50} = k_{57} = 0 \right\}$, for which $|\mathcal{K}_{\mathrm{wk}}| = 2^{448}$. For every key in $\mathcal{K}_{\mathrm{wk}}$, the imposed zero conditions on the key bytes at positions 0, 15, 22, 29, 36, 43, 50, 57 ensure that the active bytes do not generate carries that would destroy the required A/C pattern. Together with the above zero conditions on the selected constant bytes, this guarantees that the pre-whitening layer maps $\mathcal{M}_{512}^{(4)}$ to a multiset with the same integral structure.

By construction, after one forward round, $\mathcal{M}_{512}^{(4)}$ evolves into $\mathcal{M}_{512}^{(3)}$. From the previous subsection, $\mathcal{M}_{512}^{(3)}$ propagates through the next three rounds so that all 64 output bytes are balanced. Consequently, after four rounds we obtain

$$\begin{pmatrix} B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \end{pmatrix}.$$

Therefore, Kalyna-512 admits a 4-round weak-key integral distinguisher with data complexity 2^8 and weak-key space 2^{448}, for which all 64 output bytes are balanced. The detailed propagation of $\mathcal{M}_{512}^{(4)}$ through the pre-whitening layer and the subsequent four rounds of Kalyna-512 is provided in Appendix Appendix I.

5.3. 5-round integral distinguisher without pre-whitening

In this subsection, we extend the previous construction by one more round and obtain a 5-round integral distinguisher for Kalyna-512 when the pre-whitening part is omitted. Recall that, in the previous subsection, we considered the input multiset

$$\mathcal{M}_{512}^{(4)} = \begin{pmatrix} A & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & A & C \\ C & C & C & C & C & A & C & C \\ C & C & C & C & A & C & C & C \\ C & C & C & A & C & C & C & C \\ C & C & A & C & C & C & C & C \\ C & A & C & C & C & C & C & C \end{pmatrix},$$

which yields a 4-round weak-key integral distinguisher for Kalyna-512. To extend the distinguisher by one more round, we propagate the above multiset backward through one inverse round transformation

$$\mathrm{SB}^{-1} \circ \mathrm{SR}^{-1} \circ \mathrm{MC}^{-1}.$$

As before, since XOR with a fixed round subkey does not change the integral type of a byte, the inverse round-key addition does not affect the structural discussion. Applying this transformation to $\mathcal{M}_{512}^{(4)}$, we obtain a new input multiset, denoted by $\mathcal{M}_{512}^{(5)}$. Since the inverse round transformation is bijective, the multiset $\mathcal{M}_{512}^{(5)}$ also has size 2^8. At the bytewise level, this multiset may be represented schematically as

$$\mathcal{M}_{512}^{(5)} = \begin{pmatrix} A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \end{pmatrix}.$$

However, this is only a coarse bytewise description and does not capture the dependencies among the bytes. The exact multiset $\mathcal{M}_{512}^{(5)}$ is the one obtained by inverse propagation from $\mathcal{M}_{512}^{(4)}$. We now use $\mathcal{M}_{512}^{(5)}$ as the input structure for the 5-round analysis without pre-whitening. Thus, the encryption starts directly from the first round. By construction, after one forward round, $\mathcal{M}_{512}^{(5)}$ evolves into $\mathcal{M}_{512}^{(4)}$. Then, by the same 4-round propagation argument used for $\mathcal{M}_{512}^{(4)}$, all 64 output bytes are balanced after the next four rounds. Consequently, after five rounds we obtain

$$\begin{pmatrix} B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \end{pmatrix}.$$

Therefore, when the pre-whitening part is omitted, Kalyna-512 admits a 5-round integral distinguisher with data complexity 2^8, for which all 64 output bytes are balanced. The detailed propagation of $\mathcal{M}_{512}^{(5)}$ through the subsequent five rounds of Kalyna-512 is provided in Appendix Appendix J.

6. Key-recovery attacks on reduced-round Kalyna

In this section, we show how the integral distinguishers obtained in Sections 3–5 can be extended to key-recovery attacks on reduced-round variants of Kalyna. In all cases, the attack starts from a ciphertext set obtained by encrypting a chosen multiset through one or two rounds more than the corresponding distinguisher, and then partially decrypts from the ciphertext side until reaching the state for which the *Balanced* property is known.

6.1. Generic recovery principles

The key-recovery attacks considered in this paper can be classified into the following four related types:

i. Kalyna-b/b with an even number of rounds. In this case, the attacker recovers only the last-round subkey. Since the round-key addition in Kalyna is performed column-wise modulo 2^{64}, the subtraction $C_j \boxminus \widehat{K}_j^{(r)}$ cannot be handled bytewise independently because of borrow propagation. Moreover, each output byte of MC^{-1} depends on all eight bytes of the corresponding column. Hence, the natural recovery unit is one full 64-bit key column rather than a single key byte. For each ciphertext column, one guesses the corresponding 64-bit subkey column, partially decrypts the last round, and tests whether the recovered byte positions satisfy the *Balanced* property.

ii. Kalyna-b/b with an odd number of rounds. In this case, the last-round subkey is recovered in the same way. After that, the previous even-round subkey is obtained immediately from the Kalyna key-schedule relation. Thus, in odd-round attacks on Kalyna-b/b, one recovers both the last-round subkey and the preceding round subkey at essentially no additional search cost.

iii. Kalyna-$b/2b$ with an odd number of rounds. In this case, one guesses the full last-round subkey, derives the corresponding preceding-round subkey from the Kalyna key schedule, partially decrypts two rounds, and checks the *Balanced* property at the known intermediate state.

iv. Kalyna-$b/2b$ with an even number of rounds. In this case, one first guesses the full last-round subkey, partially decrypts one round, and then recovers the equivalent preceding-round key, byte by byte, by using the *Balanced* property.

We define four generic key-recovery algorithms, and the details of the corresponding key-recovery attacks are provided in Table 1.

- Algorithm 1 is used for the 4-round attacks on Kalyna-128/128, Kalyna-256/256, and the 4-round and 6-round attacks on Kalyna-512/512. In these cases, one partially decrypts only the last round and recovers the last-round subkey column by column.
- Algorithm 2 is used for the 3-round and 5-round attacks on Kalyna-b/b. Here the odd last-round subkey is recovered exactly as in the even-round case, and the preceding even-round subkey is then obtained immediately from the Kalyna key schedule.
- Algorithm 3 is used for the 5-round attacks on Kalyna-128/256 and Kalyna-256/512, where the full last-round subkey is guessed and the preceding round subkey is derived directly from the Kalyna key schedule.
- Algorithm 4 is used for the 4-round and 6-round attacks on Kalyna-128/256 and Kalyna-256/512, where one first guesses the full last-round subkey and then recovers the equivalent preceding-round key byte by byte before reconstructing the actual round subkey.

Let $m = b/64$ be the number of 64-bit columns and let $n = b/8$ be the number of bytes in the state. Let $N = |\mathcal{M}_b^{(r)}|$ be the size of one multiset. For the one-round recovery attacks, let I_j denote the set of eight byte positions recovered from the j-th ciphertext column after applying MC^{-1}, SR^{-1}, and SB^{-1}.

6.2. Complexity evaluation

We now explain how the data, time, and memory complexities of the four generic algorithms are evaluated. In all cases, two independent multisets of the same form are used in order to eliminate false candidates with overwhelming probability.

Kalyna-b/b:

Let $m = \frac{b}{64}$ be the number of 64-bit columns, and let $N = |\mathcal{M}_b^{(r)}|$ be the size of one multiset. In the one-round extensions of an r-round distinguisher, the last-round subkey is recovered column by column, so the relevant algorithms are Algorithm 1 for even target rounds and Algorithm 2 for odd target rounds. For one guessed 64-bit key column, each multiset provides eight balanced-byte conditions, so a wrong candidate survives one multiset with probability approximately $2^{-8\cdot 8} = 2^{-64}$. Hence, among the 2^{64} possible guesses, one expects about one false survivor after a single multiset, and therefore two multisets are sufficient in practice. Thus, the data complexity is $2N$, and the time complexity is approximately $m \cdot 2 \cdot 2^{64} \cdot N$ basic partial-decryption operations. The working memory complexity is negligible. If the target round is odd, then after recovering the odd last-round subkey, the preceding even-round subkey is obtained immediately from the Kalyna key schedule at essentially no additional search cost.

Algorithm 1 Generic key-recovery algorithm for Kalyna-b/b with even target round

Require: Two ciphertext multisets $\mathcal{C}_{b,1}^{(r)}$ and $\mathcal{C}_{b,2}^{(r)}$, where r is even

Ensure: The last-round subkey $K^{(r)} = (K_0^{(r)}, K_1^{(r)}, \ldots, K_{m-1}^{(r)})$

for $j = 0$ to $m - 1$ **do**
  $\mathcal{S}_j \leftarrow \emptyset$
  for all 64-bit guesses $\widehat{K}_j^{(r)}$ **do**
    flag $\leftarrow 1$
    for $\ell = 1$ to 2 **do**
      for all $i \in I_j$ **do**
        $S_i \leftarrow 0$
      end for
      for all $C \in \mathcal{C}_{b,\ell}^{(r)}$ **do**
        $U_j \leftarrow C_j \boxminus \widehat{K}_j^{(r)}$
        $W \leftarrow \mathsf{SB}^{-1} \circ \mathsf{SR}^{-1} \circ \mathsf{MC}^{-1}(U_j)$
        for all $i \in I_j$ **do**
          $S_i \leftarrow S_i \oplus W[i]$
        end for
      end for
      for all $i \in I_j$ **do**
        if $S_i \neq 0$ **then**
          flag $\leftarrow 0$
        end if
      end for
    end for
    if flag $= 1$ **then**
      $\mathcal{S}_j \leftarrow \mathcal{S}_j \cup \{\widehat{K}_j^{(r)}\}$
    end if
  end for
  Set $K_j^{(r)}$ to the unique surviving element of $\mathcal{S}_j$
end for
return $K^{(r)}$

Algorithm 2 Generic key-recovery algorithm for Kalyna-b/b with odd target round

Require: Two ciphertext multisets $\mathcal{C}_{b,1}^{(r)}$ and $\mathcal{C}_{b,2}^{(r)}$, where r is odd

Ensure: The last-round subkey $K^{(r)}$ and the previous round subkey $K^{(r-1)}$

Run Algorithm 1 on the same ciphertext multisets to recover $K^{(r)}$
Derive $K^{(r-1)}$ from $K^{(r)}$ using the Kalyna key-schedule relation
return $(K^{(r)}, K^{(r-1)})$

i. For Kalyna-128/128, all relevant distinguishers use multisets of size 2^8. Since $m = 2$, Algorithms 1 and 2 both give data complexity $2 \cdot 2^8 = 2^9$ and time complexity $2 \cdot 2 \cdot 2^{64} \cdot 2^8 = 2^{74}$. Hence, the 3-round, 4-round, and 5-round key-recovery attacks on Kalyna-128/128 all have data complexity 2^9 and time complexity 2^{74}.

ii. For Kalyna-256/256, the 3-round attack is obtained from the 2-round distinguisher and therefore uses Algorithm 2 with $N = 2^8$. Since $m = 4$, this gives data complexity is $2 \cdot 2^8 = 2^9$, and time complexity is $4 \cdot 2 \cdot 2^{64} \cdot 2^8 = 2^{75}$. The 4-round and 5-round weak-key attacks are obtained from the 3-round and 4-round weak-key distinguishers, respectively, and use Algorithms 1 and 2 with $N = 2^{16}$. Therefore, both have data complexity $2 \cdot 2^{16} = 2^{17}$ and time complexity $4 \cdot 2 \cdot 2^{64} \cdot 2^{16} = 2^{83}$.

Algorithm 3 Generic key-recovery algorithm for Kalyna-$b/2b$ with odd target round

Require: Two ciphertext multisets $\mathcal{C}_{b,1}^{(R)}$ and $\mathcal{C}_{b,2}^{(R)}$, where $R = r + 2$ is odd
Ensure: The surviving last-round subkey $K^{(R)}$ and the derived subkey $K^{(R-1)}$

$n \leftarrow b/8$
$\mathcal{S} \leftarrow \emptyset$
for all b-bit guesses $\widehat{K}^{(R)}$ **do**
    Derive $\widehat{K}^{(R-1)}$ from $\widehat{K}^{(R)}$ using the Kalyna key-schedule relation
    flag $\leftarrow 1$
    for $\ell = 1$ to 2 **do**
        for $i = 0$ to $n - 1$ **do**
            $S_i \leftarrow 0$
        end for
        for all $C \in \mathcal{C}_{b,\ell}^{(R)}$ **do**
            Partially decrypt the last two rounds of C under $\widehat{K}^{(R)}$ and $\widehat{K}^{(R-1)}$ to obtain $\widehat{X}_r$
            for $i = 0$ to $n - 1$ **do**
                $S_i \leftarrow S_i \oplus \widehat{X}_r[i]$
            end for
        end for
        for $i = 0$ to $n - 1$ **do**
            if $S_i \neq 0$ **then**
                flag $\leftarrow 0$
            end if
        end for
    end for
    if flag $= 1$ **then**
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{(\widehat{K}^{(R)}, \widehat{K}^{(R-1)})\}$
    end if
end for
return the unique surviving pair in $\mathcal{S}$

iii. For Kalyna-512/512, all relevant distinguishers use multisets of size 2^8. Since $m = 8$, Algorithm 1 yields the 4-round attack, and Algorithm 2 yields the 5-round weak-key attack. In both cases, the data complexity is $2 \cdot 2^8 = 2^9$ and the time complexity is $8 \cdot 2 \cdot 2^{64} \cdot 2^8 = 2^{76}$.

Kalyna-$b/2b$:

For Kalyna-$b/2b$, two-round extensions are considered. If the target round number is odd, then the relevant attack is the 5-round attack obtained from a 3-round distinguisher, and the corresponding procedure is given by Algorithm 3. In this case, the recovery unit is the full b-bit last-round subkey $K^{(R)}$. For each guess of $K^{(R)}$, one derives the corresponding preceding-round subkey $K^{(R-1)}$ from the Kalyna key schedule, partially decrypts two rounds, and tests the *Balanced* property on all $b/8$ byte positions of the candidate state. Since each byte-balance condition holds for a wrong guess with probability about 2^{-8}, a wrong b-bit key guess survives one multiset with probability $2^{-8\cdot(b/8)} = 2^{-b}$. Hence, among the 2^b possible guesses, one expects about one false survivor after a single multiset, and two multisets are sufficient. Therefore, if one multiset has size N, then the data complexity is $2N$, and the time complexity is approximately $2 \cdot 2^b \cdot N$ two-round partial decryptions.

If the target round number is even, then the relevant attacks are the 4-round and 6-round attacks obtained from 2-round and 4-round distinguishers, respectively, and the corresponding procedure is given by Algorithm 4. In this case, one first guesses the full b-bit last-round subkey $K^{(R)}$ and partially decrypts one round. Then one recovers the equivalent preceding-round key

Algorithm 4 Generic key-recovery algorithm for Kalyna-$b/2b$ with even target round

Require: Two ciphertext multisets $\mathcal{C}_{b,1}^{(R)}$ and $\mathcal{C}_{b,2}^{(R)}$, where $R = r + 2$ is even

Ensure: The surviving last-round subkey $K^{(R)}$, the equivalent key $\widetilde{K}^{(R-1)}$, and the derived subkeys $K^{(R-1)}$ and $K^{(R-2)}$

$n \leftarrow b/8$
$\mathcal{S} \leftarrow \emptyset$
for all b-bit guesses $\widehat{K}^{(R)}$ **do**
    flag $\leftarrow 1$
    for $i = 0$ to $n - 1$ **do**
        $\mathcal{G}_i \leftarrow \emptyset$
        for $g = 0$ to 255 **do**
            $\text{flag}_{i,g} \leftarrow 1$
            for $\ell = 1$ to 2 **do**
                $S \leftarrow 0$
                for all $C \in \mathcal{C}_{b,\ell}^{(R)}$ **do**
                    Partially decrypt the last round of C under $\widehat{K}^{(R)}$ to obtain $\widehat{X}_{R-1}$
                    $Y \leftarrow \mathsf{SR}^{-1}\Big(\mathsf{MC}^{-1}(\widehat{X}_{R-1})\Big)$
                    $S \leftarrow S \oplus \mathsf{SB}^{-1}(Y[i] \oplus g)$
                end for
                if $S \neq 0$ **then**
                    $\text{flag}_{i,g} \leftarrow 0$
                end if
            end for
            if $\text{flag}_{i,g} = 1$ **then**
                $\mathcal{G}_i \leftarrow \mathcal{G}_i \cup \{g\}$
            end if
        end for
        if $|\mathcal{G}_i| \neq 1$ **then**
            flag $\leftarrow 0$
        else
            $\widetilde{K}^{(R-1)}[i] \leftarrow$ the unique element of $\mathcal{G}_i$
        end if
    end for
    if flag $= 1$ **then**
        $K^{(R-1)} \leftarrow \mathsf{MC}\Big(\mathsf{SR}(\widetilde{K}^{(R-1)})\Big)$
        Derive $K^{(R-2)}$ from $K^{(R-1)}$ using the Kalyna key-schedule relation
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{(\widehat{K}^{(R)}, \widetilde{K}^{(R-1)}, K^{(R-1)}, K^{(R-2)})\}$
    end if
end for
return the unique surviving tuple in $\mathcal{S}$

$\widetilde{K}^{(R-1)}$ byte by byte. For each fixed guess of $K^{(R)}$, recovering all $b/8$ bytes of $\widetilde{K}^{(R-1)}$ costs approximately $\frac{b}{8} \cdot 2 \cdot 2^8 \cdot N$ basic operations. Multiplying this by the 2^b possible guesses for $K^{(R)}$, we obtain total data complexity $2N$ and time complexity $2^b \cdot \frac{b}{8} \cdot 2 \cdot 2^8 \cdot N$. After $\widetilde{K}^{(R-1)}$ has been recovered, one reconstructs the actual round subkey $K^{(R-1)}$, and the preceding subkey $K^{(R-2)}$ is then derived immediately from the Kalyna key schedule. The working memory complexity is again negligible.

i. For Kalyna-128/256, the 4-round and 6-round attacks are obtained from the 2-round and 4-round distinguishers, respectively, and therefore use Algorithm 4. Since both distin-

guishers use multisets of size 2^8, we have $b = 128$ and $N = 2^8$, so both attacks have data complexity $2 \cdot 2^8 = 2^9$ and time complexity $2^{128} \cdot \frac{128}{8} \cdot 2 \cdot 2^8 \cdot 2^8 = 2^{149}$. The 5-round attack is obtained from the 3-round weak-key distinguisher and uses Algorithm 3. Here again $b = 128$ and $N = 2^8$, so the data complexity is $2 \cdot 2^8 = 2^9$ and the time complexity is $2 \cdot 2^{128} \cdot 2^8 = 2^{137}$.

ii. For Kalyna-256/512, the 4-round attack is obtained from the 2-round distinguisher and uses Algorithm 4 with $b = 256$ and $N = 2^8$. Thus, its data complexity is $2 \cdot 2^8 = 2^9$ and its time complexity is $2^{256} \cdot \frac{256}{8} \cdot 2 \cdot 2^8 \cdot 2^8 = 2^{278}$. The 5-round attack is obtained from the 3-round weak-key distinguisher and uses Algorithm 3 with $b = 256$ and $N = 2^{16}$. Hence, its data complexity is $2 \cdot 2^{16} = 2^{17}$ and its time complexity is $2 \cdot 2^{256} \cdot 2^{16} = 2^{273}$. Finally, the 6-round attack is obtained from the 4-round weak-key distinguisher and uses Algorithm 4 with $b = 256$ and $N = 2^{16}$. Therefore, its data complexity is $2 \cdot 2^{16} = 2^{17}$ and its time complexity is $2^{256} \cdot \frac{256}{8} \cdot 2 \cdot 2^8 \cdot 2^{16} = 2^{286}$.

Thus, all complexity values reported in the subsequent subsections are obtained directly by substituting the appropriate multiset size into the above formulae; the corresponding complexity details are summarized in Table 1.

7. Discussion and limitations

7.1. Discussion

The results obtained in this paper provide a unified integral view of the three standard Kalyna variants with block sizes 128, 256, and 512 bits. A common feature in all our constructions is that the propagation of integral properties is determined by the interaction between the ShiftRows operation, the MDS-based linear diffusion layer, and the column-wise modular addition used in the pre-whitening part. In particular, carefully chosen active-byte positions allow the linear layer to spread the activity rapidly, while the modular additions may either preserve or disturb the required bytewise structure depending on carry propagation inside each 64-bit column.

From this perspective, the main contribution of the paper is not only the explicit reduced-round distinguishers for Kalyna-128, Kalyna-256, and Kalyna-512, but also the structural understanding of how integral properties can still be controlled in an SPN cipher that combines AES-like diffusion with addition modulo 2^{64}. The obtained weak-key and no-pre-whitening extensions further illustrate that the modular pre-whitening layer is the main obstacle to extending these distinguishers to more rounds. Moreover, the distinguishers derived in Sections 3 to 5 naturally support the key-recovery attacks considered in Section 6. In particular, the *Balanced* property at the penultimate state allows partial decryption of the last round from the ciphertext side and provides filtering conditions for candidate subkeys. Since Kalyna applies the column-wise modulo of key addition 2^{64}, the natural recovery unit is a full 64-bit key column rather than an individual byte. This makes the recovery phase structurally different from the usual bytewise last-round guessing strategy used for many XOR-based SPN ciphers.

7.2. Limitations

The results obtained in this paper should be interpreted with certain limitations. First, several of the longer distinguishers are derived only under additional assumptions. More precisely, some of them hold in a weak-key setting, while others are obtained only when the pre-whitening operation is omitted. Therefore, these results do not directly translate into attacks on the full standard versions of Kalyna. Second, the present analysis is restricted to reduced-round variants. Although the obtained distinguishers and key-recovery extensions help clarify the integral behavior of Kalyna, they do not threaten the security of the full-round cipher. Rather, they should be viewed as structural results that improve the understanding of how integral properties

propagate through the round function and how the pre-whitening layer affects this propagation. A further limitation is that the standard bytewise notation using the symbols A, B, C, and U becomes less informative after several rounds of backward extension. In some cases, the corresponding input multiset remains well defined through exact inverse-round propagation, but it can no longer be conveniently described by a simple bytewise pattern. Thus, the symbolic notation should be regarded as a coarse description rather than a complete representation of all dependencies inside the state.

Finally, in the key-recovery phase, the presence of column-wise addition modulo 2^{64} in the whitening layers makes the recovery process less flexible than in XOR-only SPN ciphers. In particular, the natural recovery unit is a whole 64-bit column subkey, which may increase the complexity of extending the distinguishers into practical attacks on more rounds. This shows that the main difficulty in Kalyna is not only the diffusion layer itself, but also the interaction between diffusion and modular addition.

8. Conclusions

In this paper, we investigated integral cryptanalysis of the Ukrainian block cipher Kalyna and studied the propagation of integral properties for its three standard block sizes, namely Kalyna-128, Kalyna-256, and Kalyna-512. By combining carefully chosen input multisets with backward extension through inverse round transformations, we derived several reduced-round integral distinguishers and analyzed the effect of the pre-whitening layer on their construction. We also showed how these distinguishers can be extended to key-recovery attacks on reduced-round variants of Kalyna, where the *Balanced* property of the partially decrypted state provides filtering conditions for candidate last-round subkeys. Since Kalyna uses column-wise addition modulo 2^{64} in its whitening layers, the natural recovery unit is a whole 64-bit key column rather than an individual byte.

For Kalyna-128, we obtained a 2-round integral distinguisher, a 3-round weak-key integral distinguisher, and a 4-round integral distinguisher without pre-whitening. For Kalyna-256, we derived a 2-round integral distinguisher together with 3-round and 4-round weak-key integral distinguishers. For Kalyna-512, we presented a 3-round integral distinguisher, a 4-round weak-key integral distinguisher, and a 5-round integral distinguisher without pre-whitening. In all these cases, the resulting output bytes satisfy the *Balanced* property.

Our analysis highlights the central role of the modular addition used in the pre-whitening layer. In particular, carry propagation inside 64-bit columns is the main obstacle to extending bytewise integral structures, which explains why weak-key restrictions or the omission of pre-whitening become necessary in several of the longer-round constructions. Overall, these results provide a unified view of the reduced-round integral behavior of Kalyna and improve the understanding of how integral properties interact with an SPN design that combines AES-like diffusion with modular arithmetic. Several directions for future work remain open. One natural continuation is to combine the present manual analysis with automated techniques such as the division property or MILP-based search in order to investigate whether stronger or longer integral distinguishers can be obtained for Kalyna.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data and material

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Funding

The work of Nitish Kumar was supported by the Ministry of Education (MoE), Government of India.

Authors' contributions

All authors contributed equally to this work.

Acknowledgements

Not applicable.

References

[1] X. Lai. *Higher order derivatives and differential cryptanalysis*, in Communications and Cryptography: Two Sides of One Tapestry, R. E. Blahut, D. J. Costello Jr., U. Maurer, and T. Mittelholzer (eds.), The Springer International Series in Engineering and Computer Science, 276 (1994), 227–233.

[2] L. R. Knudsen. *Truncated and higher order differentials*, in Fast Software Encryption, Lecture Notes in Computer Science, 1008 (1995), 196–211.

[3] J. Daemen, L. R. Knudsen, and V. Rijmen. *The block cipher Square*, in Fast Software Encryption, Lecture Notes in Computer Science, 1267 (1997), 149–165.

[4] L. R. Knudsen and D. Wagner. *Integral cryptanalysis*, in Fast Software Encryption, Lecture Notes in Computer Science, 2365 (2002), 112–127.

[5] S. Lucks. *The saturation attack — a bait for Twofish*, in Fast Software Encryption, Lecture Notes in Computer Science, 2355 (2001), 1–15.

[6] Y. Todo. *Structural evaluation by generalized integral property*, in Advances in Cryptology – EUROCRYPT 2015, Part I, Lecture Notes in Computer Science, 9056 (2015), 287–314.

[7] Y. Todo and M. Morii. *Bit-based division property and application to Simon family*, in Fast Software Encryption, Lecture Notes in Computer Science, 9783 (2016), 357–377.

[8] Y. Todo. *Integral cryptanalysis on full MISTY1*, Journal of Cryptology, 30:3 (2017), 920–959.

[9] Z. Xiang, W. Zhang, Z. Bao, and D. Lin. *Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers*, in Advances in Cryptology – ASIACRYPT 2016, Part II, Lecture Notes in Computer Science, 10031 (2016), 648–678.

[10] L. Lin and W. Wu. *Improved meet-in-the-middle attacks on reduced-round Kalyna-128/256 and Kalyna-256/512*, Designs, Codes and Cryptography, 86(4) (2018), 721–741.

[11] R. AlTawy, A. Abdelkhalek, and A. M. Youssef. *A meet-in-the-middle attack on reduced-round Kalyna-b/2b*, IEICE Transactions on Information and Systems, E99.D(4) (2016), 1246–1250.

[12] Akshima, D. Chang, M. Ghosh, A. Goel, and S. K. Sanadhya. *Single key recovery attacks on 9-round Kalyna-128/256 and Kalyna-256/512*, Information Security and Cryptology – ICISC 2015, Lecture Notes in Computer Science, 9558 (2016), 119–135.

[13] M. ElSheikh and A. M. Youssef. *On MILP-based automatic search for bit-based division property for ciphers with (large) linear layers*, in Progress' in Cryptology – ACISP 2021, Lecture Notes in Computer Science, 12727 (2021), 111–131.

[14] R. Oliynykov, I. Gorbenko, O. Kazymyrov, V. Ruzhentsev, O. Kuznetsov, Y. Gorbenko, O. Dyrda, V. Dolgov, A. Pushkaryov, R. Mordvinov, and D. Kaidalov. *A new encryption standard of Ukraine: The Kalyna block cipher*, Cryptology ePrint Archive, Report 2015/650, 2015.

Appendix A. Kalyna S-boxes and linear layer

In this appendix, we collect the basic nonlinear and linear components of the Kalyna block cipher for the convenience of the reader. More precisely, we give the four 8-bit S-boxes S_0, S_1, S_2, and S_3, the corresponding inverse S-boxes $S_0^{-1}, S_1^{-1}, S_2^{-1}$, and S_3^{-1}, and the MDS matrix used in the linear diffusion layer together with its inverse. These components are used repeatedly in the specification and cryptanalytic analysis of Kalyna. The forward round of Kalyna employs four 8-bit substitution boxes, denoted by S_0, S_1, S_2, and S_3, which are listed below.

A8	43	5F	06	6B	75	6C	59	71	DF	87	95	17	F0	D8	09
6D	F3	1D	CB	C9	4D	2C	AF	79	E0	97	FD	6F	4B	45	39
3E	DD	A3	4F	B4	B6	9A	0E	1F	BF	15	E1	49	D2	93	C6
92	72	9E	61	D1	63	FA	EE	F4	19	D5	AD	58	A4	BB	A1
DC	F2	83	37	42	E4	7A	32	9C	CC	AB	4A	8F	6E	04	27
2E	E7	E2	5A	96	16	23	2B	C2	65	66	0F	BC	A9	47	41
34	48	FC	B7	6A	88	A5	53	86	F9	5B	DB	38	7B	C3	1E
22	33	24	28	36	C7	B2	3B	8E	77	BA	F5	14	9F	08	55
9B	4C	FE	60	5C	DA	18	46	CD	7D	21	B0	3F	1B	89	FF
EB	84	69	3A	9D	D7	D3	70	67	40	B5	DE	5D	30	91	B1
78	11	01	E5	00	68	98	A0	C5	02	A6	74	2D	0B	A2	76
B3	BE	CE	BD	AE	E9	8A	31	1C	EC	F1	99	94	AA	F6	26
2F	EF	E8	8C	35	03	D4	7F	FB	05	C1	5E	90	20	3D	82
F7	EA	0A	0D	7E	F8	50	1A	C4	07	57	B8	3C	62	E3	C8
AC	52	64	10	D0	D9	13	0C	12	29	51	B9	CF	D6	73	8D
81	54	C0	ED	4E	44	A7	2A	85	25	E6	CA	7C	8B	56	80

S-box S_0

CE	BB	EB	92	EA	CB	13	C1	E9	3A	D6	B2	D2	90	17	F8
42	15	56	B4	65	1C	88	43	C5	5C	36	BA	F5	57	67	8D
31	F6	64	58	9E	F4	22	AA	75	0F	02	B1	DF	6D	73	4D
7C	26	2E	F7	08	5D	44	3E	9F	14	C8	AE	54	10	D8	BC
1A	6B	69	F3	BD	33	AB	FA	D1	9B	68	4E	16	95	91	EE
4C	63	8E	5B	CC	3C	19	A1	81	49	7B	D9	6F	37	60	CA
E7	2B	48	FD	96	45	FC	41	12	0D	79	E5	89	8C	E3	20
30	DC	B7	6C	4A	B5	3F	97	D4	62	2D	06	A4	A5	83	5F
2A	DA	C9	00	7E	A2	55	BF	11	D5	9C	CF	0E	0A	3D	51
7D	93	1B	FE	C4	47	09	86	0B	8F	9D	6A	07	B9	B0	98
18	32	71	4B	EF	3B	70	A0	E4	40	FF	C3	A9	E6	78	F9
8B	46	80	1E	38	E1	B8	A8	E0	0C	23	76	1D	25	24	05
F1	6E	94	28	9A	84	E8	A3	4F	77	D3	85	E2	52	F2	82
50	7A	2F	74	53	B3	61	AF	39	35	DE	CD	1F	99	AC	AD
72	2C	DD	D0	87	BE	5E	A6	EC	04	C6	03	34	FB	DB	59
B6	C2	01	F0	5A	ED	A7	66	21	7F	8A	27	C7	C0	29	D7

S-box S_1

93	$D9$	$9A$	$B5$	98	22	45	FC	BA	$6A$	DF	02	$9F$	DC	51	59
$4A$	17	$2B$	$C2$	94	$F4$	BB	$A3$	62	$E4$	71	$D4$	CD	70	16	$E1$
49	$3C$	$C0$	$D8$	$5C$	$9B$	AD	85	53	$A1$	$7A$	$C8$	$2D$	$E0$	$D1$	72
$A6$	$2C$	$C4$	$E3$	76	78	$B7$	$B4$	09	$3B$	$0E$	41	$4C$	DE	$B2$	90
25	$A5$	$D7$	03	11	00	$C3$	$2E$	92	EF	$4E$	12	$9D$	$7D$	CB	35
10	$D5$	$4F$	$9E$	$4D$	$A9$	55	$C6$	$D0$	$7B$	18	97	$D3$	36	$E6$	48
56	81	$8F$	77	CC	$9C$	$B9$	$E2$	AC	$B8$	$2F$	15	$A4$	$7C$	DA	38
$1E$	$0B$	05	$D6$	14	$6E$	$6C$	$7E$	66	FD	$B1$	$E5$	60	AF	$5E$	33
87	$C9$	$F0$	$5D$	$6D$	$3F$	88	$8D$	$C7$	$F7$	$1D$	$E9$	EC	ED	80	29
27	CF	99	$A8$	50	$0F$	37	24	28	30	95	$D2$	$3E$	$5B$	40	83
$B3$	69	57	$1F$	07	$1C$	$8A$	BC	20	EB	CE	$8E$	AB	EE	31	$A2$
73	$F9$	CA	$3A$	$1A$	FB	$0D$	$C1$	FE	FA	$F2$	$6F$	BD	96	DD	43
52	$B6$	08	$F3$	AE	BE	19	89	32	26	$B0$	EA	$4B$	64	84	82
$6B$	$F5$	79	BF	01	$5F$	75	63	$1B$	23	$3D$	68	$2A$	65	$E8$	91
$F6$	FF	13	58	$F1$	47	$0A$	$7F$	$C5$	$A7$	$E7$	61	$5A$	06	46	44
42	04	$A0$	DB	39	86	54	AA	$8C$	34	21	$8B$	$F8$	$0C$	74	67

S-box S_2

68	$8D$	CA	$4D$	73	$4B$	$4E$	$2A$	$D4$	52	26	$B3$	54	$1E$	19	$1F$
22	03	46	$3D$	$2D$	$4A$	53	83	13	$8A$	$B7$	$D5$	25	79	$F5$	BD
58	$2F$	$0D$	02	ED	51	$9E$	11	$F2$	$3E$	55	$5E$	$D1$	16	$3C$	66
70	$5D$	$F3$	45	40	CC	$E8$	94	56	08	CE	$1A$	$3A$	$D2$	$E1$	DF
$B5$	38	$6E$	$0E$	$E5$	$F4$	$F9$	86	$E9$	$4F$	$D6$	85	23	CF	32	99
31	14	AE	EE	$C8$	48	$D3$	30	$A1$	92	41	$B1$	18	$C4$	$2C$	71
72	44	15	FD	37	BE	$5F$	AA	$9B$	88	$D8$	AB	89	$9C$	FA	60
EA	BC	62	$0C$	24	$A6$	$A8$	EC	67	20	DB	$7C$	28	DD	AC	$5B$
34	$7E$	10	$F1$	$7B$	$8F$	63	$A0$	05	$9A$	43	77	21	BF	27	09
$C3$	$9F$	$B6$	$D7$	29	$C2$	EB	$C0$	$A4$	$8B$	$8C$	$1D$	FB	FF	$C1$	$B2$
97	$2E$	$F8$	65	$F6$	75	07	04	49	33	$E4$	$D9$	$B9$	$D0$	42	$C7$
$6C$	90	00	$8E$	$6F$	50	01	$C5$	DA	47	$3F$	CD	69	$A2$	$E2$	$7A$
$A7$	$C6$	93	$0F$	$0A$	06	$E6$	$2B$	96	$A3$	$1C$	AF	$6A$	12	84	39
$E7$	$B0$	82	$F7$	FE	$9D$	87	$5C$	81	35	DE	$B4$	$A5$	FC	80	EF
CB	BB	$6B$	76	BA	$5A$	$7D$	78	$0B$	95	$E3$	AD	74	98	$3B$	36
64	$6D$	DC	$F0$	59	$A9$	$4C$	17	$7F$	91	$B8$	$C9$	57	$1B$	$E0$	61

S-box S_3

The inverse round of Kalyna employs four 8-bit inverse substitution boxes, denoted by $S_0^{-1}, S_1^{-1}, S_2^{-1}$, and S_3^{-1}, which are listed below.

$A4$	$A2$	$A9$	$C5$	$4E$	$C9$	03	$D9$	$7E$	$0F$	$D2$	AD	$E7$	$D3$	27	$5B$
$E3$	$A1$	$E8$	$E6$	$7C$	$2A$	55	$0C$	86	39	$D7$	$8D$	$B8$	12	$6F$	28
CD	$8A$	70	56	72	$F9$	BF	$4F$	73	$E9$	$F7$	57	16	AC	50	$C0$
$9D$	$B7$	47	71	60	$C4$	74	43	$6C$	$1F$	93	77	DC	CE	20	$8C$
99	$5F$	44	01	$F5$	$1E$	87	$5E$	61	$2C$	$4B$	$1D$	81	15	$F4$	23
$D6$	EA	$E1$	67	$F1$	$7F$	FE	DA	$3C$	07	53	$6A$	84	$9C$	CB	02
83	33	DD	35	$E2$	59	$5A$	98	$A5$	92	64	04	06	10	$4D$	$1C$
97	08	31	EE	AB	05	AF	79	$A0$	18	46	$6D$	FC	89	$D4$	$C7$
FF	$F0$	CF	42	91	$F8$	68	$0A$	65	$8E$	$B6$	FD	$C3$	EF	78	$4C$
CC	$9E$	30	$2E$	BC	$0B$	54	$1A$	$A6$	BB	26	80	48	94	32	$7D$
$A7$	$3F$	AE	22	$3D$	66	AA	$F6$	00	$5D$	BD	$4A$	$E0$	$3B$	$B4$	17
$8B$	$9F$	76	$B0$	24	$9A$	25	63	DB	EB	$7A$	$3E$	$5C$	$B3$	$B1$	29
$F2$	CA	58	$6E$	$D8$	$A8$	$2F$	75	DF	14	FB	13	49	88	$B2$	EC
$E4$	34	$2D$	96	$C6$	$3A$	ED	95	$0E$	$E5$	85	$6B$	40	21	$9B$	09
19	$2B$	52	DE	45	$A3$	FA	51	$C2$	$B5$	$D1$	90	$B9$	$F3$	37	$C1$
$0D$	BA	41	11	38	$7B$	BE	$D0$	$D5$	69	36	$C8$	62	$1B$	82	$8F$

Inverse S-box S_0^{-1}

83	$F2$	$2A$	EB	$E9$	BF	$7B$	$9C$	34	96	$8D$	98	$B9$	69	$8C$	29
$3D$	88	68	06	39	11	$4C$	$0E$	$A0$	56	40	92	15	BC	$B3$	DC
$6F$	$F8$	26	BA	BE	BD	31	FB	$C3$	FE	80	61	$E1$	$7A$	32	$D2$
70	20	$A1$	45	EC	$D9$	$1A$	$5D$	$B4$	$D8$	09	$A5$	55	$8E$	37	76
$A9$	67	10	17	36	65	$B1$	95	62	59	74	$A3$	50	$2F$	$4B$	$C8$
$D0$	$8F$	CD	$D4$	$3C$	86	12	$1D$	23	EF	$F4$	53	19	35	$E6$	$7F$
$5E$	$D6$	79	51	22	14	$F7$	$1E$	$4A$	42	$9B$	41	73	$2D$	$C1$	$5C$
$A6$	$A2$	$E0$	$2E$	$D3$	28	BB	$C9$	AE	$6A$	$D1$	$5A$	30	90	84	$F9$
$B2$	58	CF	$7E$	$C5$	CB	97	$E4$	16	$6C$	FA	$B0$	$6D$	$1F$	52	99
$0D$	$4E$	03	91	$C2$	$4D$	64	77	$9F$	DD	$C4$	49	$8A$	$9A$	24	38
$A7$	57	85	$C7$	$7C$	$7D$	$E7$	$F6$	$B7$	AC	27	46	DE	DF	$3B$	$D7$
$9E$	$2B$	$0B$	$D5$	13	75	$F0$	72	$B6$	$9D$	$1B$	01	$3F$	44	$E5$	87
FD	07	$F1$	AB	94	18	EA	FC	$3A$	82	$5F$	05	54	DB	00	$8B$
$E3$	48	$0C$	CA	78	89	$0A$	FF	$3E$	$5B$	81	EE	71	$E2$	DA	$2C$
$B8$	$B5$	CC	$6E$	$A8$	$6B$	AD	60	$C6$	08	04	02	$E8$	$F5$	$4F$	$A4$
$F3$	$C0$	CE	43	25	$1C$	21	33	$0F$	AF	47	ED	66	63	93	AA

Inverse S-box S_1^{-1}

45	$D4$	$0B$	43	$F1$	72	ED	$A4$	$C2$	38	$E6$	71	FD	$B6$	$3A$	95
50	44	$4B$	$E2$	74	$6B$	$1E$	11	$5A$	$C6$	$B4$	$D8$	$A5$	$8A$	70	$A3$
$A8$	FA	05	$D9$	97	40	$C9$	90	98	$8F$	DC	12	31	$2C$	47	$6A$
99	AE	$C8$	$7F$	$F9$	$4F$	$5D$	96	$6F$	$F4$	$B3$	39	21	DA	$9C$	85
$9E$	$3B$	$F0$	BF	EF	06	EE	$E5$	$5F$	20	10	CC	$3C$	54	$4A$	52
94	$0E$	$C0$	28	$F6$	56	60	$A2$	$E3$	$0F$	EC	$9D$	24	83	$7E$	$D5$
$7C$	EB	18	$D7$	CD	DD	78	FF	DB	$A1$	09	$D0$	76	84	75	BB
$1D$	$1A$	$2F$	$B0$	FE	$D6$	34	63	35	$D2$	$2A$	59	$6D$	$4D$	77	$E7$
$8E$	61	CF	$9F$	CE	27	$F5$	80	86	$C7$	$A6$	FB	$F8$	87	AB	62
$3F$	DF	48	00	14	$9A$	BD	$5B$	04	92	02	25	65	$4C$	53	$0C$
$F2$	29	AF	17	$6C$	41	30	$E9$	93	55	$F7$	AC	68	26	$C4$	$7D$
CA	$7A$	$3E$	$A0$	37	03	$C1$	36	69	66	08	16	$A7$	BC	$C5$	$D3$
22	$B7$	13	46	32	$E8$	57	88	$2B$	81	$B2$	$4E$	64	$1C$	AA	91
58	$2E$	$9B$	$5C$	$1B$	51	73	42	23	01	$6E$	$F3$	$0D$	BE	$3D$	$0A$
$2D$	$1F$	67	33	19	$7B$	$5E$	EA	DE	$8B$	CB	$A9$	$8C$	$8D$	AD	49
82	$E4$	BA	$C3$	15	$D1$	$E0$	89	FC	$B1$	$B9$	$B5$	07	79	$B8$	$E1$

Inverse S-box S_2^{-1}

$B2$	$B6$	23	11	$A7$	88	$C5$	$A6$	39	$8F$	$C4$	$E8$	73	22	43	$C3$
82	27	CD	18	51	62	$2D$	$F7$	$5C$	$0E$	$3B$	FD	CA	$9B$	$0D$	$0F$
79	$8C$	10	$4C$	74	$1C$	$0A$	$8E$	$7C$	94	07	$C7$	$5E$	14	$A1$	21
57	50	$4E$	$A9$	80	$D9$	EF	64	41	CF	$3C$	EE	$2E$	13	29	BA
34	$5A$	AE	$8A$	61	33	12	$B9$	55	$A8$	15	05	$F6$	03	06	49
$B5$	25	09	16	$0C$	$2A$	38	FC	20	$F4$	$E5$	$7F$	$D7$	31	$2B$	66
$6F$	FF	72	86	$F0$	$A3$	$2F$	78	00	BC	CC	$E2$	$B0$	$F1$	42	$B4$
30	$5F$	60	04	EC	$A5$	$E3$	$8B$	$E7$	$1D$	BF	84	$7B$	$E6$	81	$F8$
DE	$D8$	$D2$	17	CE	$4B$	47	$D6$	69	$6C$	19	99	$9A$	01	$B3$	85
$B1$	$F9$	59	$C2$	37	$E9$	$C8$	$A0$	ED	$4F$	89	68	$6D$	$D5$	26	91
87	58	BD	$C9$	98	DC	75	$C0$	76	$F5$	67	$6B$	$7E$	EB	52	CB
$D1$	$5B$	$9F$	$0B$	DB	40	92	$1A$	FA	AC	$E4$	$E1$	71	$1F$	65	$8D$
97	$9E$	95	90	$5D$	$B7$	$C1$	AF	54	FB	02	$E0$	35	BB	$3A$	$4D$
AD	$2C$	$3D$	56	08	$1B$	$4A$	93	$6A$	AB	$B8$	$7A$	$F2$	$7D$	DA	$3F$
FE	$3E$	BE	EA	AA	44	$C6$	$D0$	36	48	70	96	77	24	53	DF
$F3$	83	28	32	45	$1E$	$A4$	$D3$	$A2$	46	$6E$	$9C$	DD	63	$D4$	$9D$

Inverse S-box S_3^{-1}

The forward linear diffusion layer of Kalyna is defined by an 8×8 MDS matrix, which is given below.

$$M = \begin{pmatrix} 01 & 01 & 05 & 01 & 08 & 06 & 07 & 04 \\ 04 & 01 & 01 & 05 & 01 & 08 & 06 & 07 \\ 07 & 04 & 01 & 01 & 05 & 01 & 08 & 06 \\ 06 & 07 & 04 & 01 & 01 & 05 & 01 & 08 \\ 08 & 06 & 07 & 04 & 01 & 01 & 05 & 01 \\ 01 & 08 & 06 & 07 & 04 & 01 & 01 & 05 \\ 05 & 01 & 08 & 06 & 07 & 04 & 01 & 01 \\ 01 & 05 & 01 & 08 & 06 & 07 & 04 & 01 \end{pmatrix}$$

The inverse linear diffusion layer of Kalyna is defined by the corresponding inverse 8×8 MDS matrix, which is given below.

$$M^{-1} = \begin{pmatrix} AD & 95 & 76 & A8 & 2F & 49 & D7 & CA \\ 95 & 76 & A8 & 2F & 49 & D7 & CA & AD \\ 76 & A8 & 2F & 49 & D7 & CA & AD & 95 \\ A8 & 2F & 49 & D7 & CA & AD & 95 & 76 \\ 2F & 49 & D7 & CA & AD & 95 & 76 & A8 \\ 49 & D7 & CA & AD & 95 & 76 & A8 & 2F \\ D7 & CA & AD & 95 & 76 & A8 & 2F & 49 \\ CA & AD & 95 & 76 & A8 & 2F & 49 & D7 \end{pmatrix}$$

Appendix B. Propagation of the 2-round integral distinguisherfor Kalyna-128

In this appendix, we present the detailed propagation of the multiset $\mathcal{M}_{128}^{(2)}$ used in the 2-round integral distinguisher for Kalyna-128. Recall that

$$\mathcal{M}_{128}^{(2)} = \begin{pmatrix} C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ A & C \end{pmatrix}.$$

Here, the active byte is located at the 7th byte of the first column, while all remaining bytes are constant. As discussed in the main text, the pre-whitening operation of Kalyna-128 does not affect the integral structure of $\mathcal{M}_{128}^{(2)}$. Hence, after the pre-whitening step, the state remains unchanged in terms of $A/C/B/U$ properties.

Pre-whitening and Round 1

$$\mathcal{M}_{128}^{(2)} \xrightarrow{\boxplus K^{pw}} \begin{pmatrix} C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ A & C \end{pmatrix} \xrightarrow{SB} \begin{pmatrix} C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ A & C \end{pmatrix} \xrightarrow{SR} \begin{pmatrix} C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & A \end{pmatrix} \xrightarrow{MC} \begin{pmatrix} C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \end{pmatrix} \xrightarrow{\oplus K_1} \begin{pmatrix} C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \end{pmatrix}$$

In the first round, the substitution layer preserves the active byte, since the S-box is bijective. The ShiftRows operation moves the active byte from the first column to the second column because row 7 is shifted by one position. Then, the linear layer diffuses the activity to all bytes of the second column. The round-key XOR preserves the integral type of each byte.

Round 2.

$$\begin{pmatrix} C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \end{pmatrix} \xrightarrow{SB} \begin{pmatrix} C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \end{pmatrix} \xrightarrow{SR} \begin{pmatrix} C & A \\ C & A \\ C & A \\ C & A \\ A & C \\ A & C \\ A & C \\ A & C \end{pmatrix} \xrightarrow{MC} \begin{pmatrix} B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \end{pmatrix} \xrightarrow{\oplus K_2} \begin{pmatrix} B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \end{pmatrix}$$

In the second round, the substitution layer again preserves the integral type of each byte. After the ShiftRows operation, the active bytes in rows $4, 5, 6, 7$ move to the first column, while those in rows $0, 1, 2, 3$ remain in the second column. Hence, each column contains four active bytes and four constant bytes. After applying the linear layer, all 16 output bytes satisfy the *Balanced* property. The final round-key addition preserves this balanced structure. Therefore, after two rounds of Kalyna-128, all 16 output bytes are balanced, which proves the 2-round integral distinguisher.

Appendix C. Propagation of the 3-round weak-key integral distinguisher for Kalyna-128

In this appendix, we present the detailed propagation of the multiset $\mathcal{M}_{128}^{(3)}$ used in the 3-round weak-key integral distinguisher for Kalyna-128. We start from the multiset

$$\mathcal{M}_{128}^{(3)} = \begin{pmatrix} A & C \\ A & C \\ A & C \\ A & C \\ C & A \\ C & A \\ C & A \\ C & A \end{pmatrix}.$$

We include the pre-whitening layer and consider the weak-key space $\mathcal{K}_{\text{wk}} = \Big\{ K^{\text{pw}} = (k_0, \ldots, k_{15}) \in (\mathbb{F}_2^8)^{16} \;\Big|\; k_0 = k_1 = k_2 = k_3 = k_{12} = k_{13} = k_{14} = k_{15} = 0 \Big\}$. Thus, $|\mathcal{K}_{\text{wk}}| = 2^{64}$. Moreover, the fixed constants in the plaintext structure are chosen such that $c_8 = c_9 = c_{10} = c_{11} = 0$. Under these conditions, the pre-whitening addition modulo 2^{64} preserves the bytewise A/C structure of $\mathcal{M}_{128}^{(3)}$. Indeed, in the first 64-bit column, the condition $k_0 = k_1 = k_2 = k_3 = 0$ leaves the active part unchanged and prevents carries from propagating into the upper constant part. In the second 64-bit column, the choice $c_8 = c_9 = c_{10} = c_{11} = 0$ makes the lower 32-bit part equal to zero before pre-whitening. Hence, after addition with the fixed bytes k_8, k_9, k_{10}, k_{11}, this part remains constant and, in particular, no carry is generated from byte 11 into byte 12. Furthermore, the condition $k_{12} = k_{13} = k_{14} = k_{15} = 0$ ensures that the active bytes $12, 13, 14, 15$ are not modified by the key addition itself, so no key-dependent carry propagates inside the active part. Therefore, the second column preserves the same bytewise structure after pre-whitening: bytes $8, 9, 10, 11$ remain constant, while bytes $12, 13, 14, 15$ remain active.

Pre-whitening.

$\mathcal{M}_{128}^{(3)} \xrightarrow{\boxplus K^{\text{pw}}} \mathcal{M}_{128}^{(3)}$ for all $K^{\text{pw}} \in \mathcal{K}_{\text{wk}}$, in terms of integral structure.

Round 1.

$$\begin{pmatrix} A & C \\ A & C \\ A & C \\ A & C \\ C & A \\ C & A \\ C & A \\ C & A \end{pmatrix} \xrightarrow{SB} \begin{pmatrix} A & C \\ A & C \\ A & C \\ A & C \\ C & A \\ C & A \\ C & A \\ C & A \end{pmatrix} \xrightarrow{SR} \begin{pmatrix} A & C \\ A & C \\ A & C \\ A & C \\ A & C \\ A & C \\ A & C \\ A & C \end{pmatrix} \xrightarrow{MC} \begin{pmatrix} C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ A & C \end{pmatrix} \xrightarrow{\oplus K_1} \begin{pmatrix} C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ A & C \end{pmatrix}$$

Round 2.

$$\begin{pmatrix} C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ A & C \end{pmatrix} \xrightarrow{SB} \begin{pmatrix} C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ A & C \end{pmatrix} \xrightarrow{SR} \begin{pmatrix} C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & A \end{pmatrix} \xrightarrow{MC} \begin{pmatrix} C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \end{pmatrix} \xrightarrow{\oplus K_2} \begin{pmatrix} C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \end{pmatrix}$$

Round 3.

$$\begin{pmatrix} C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \end{pmatrix} \xrightarrow{SB} \begin{pmatrix} C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \\ C & A \end{pmatrix} \xrightarrow{SR} \begin{pmatrix} C & A \\ C & A \\ C & A \\ C & A \\ A & C \\ A & C \\ A & C \\ A & C \end{pmatrix} \xrightarrow{MC} \begin{pmatrix} B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \end{pmatrix} \xrightarrow{\oplus K_3} \begin{pmatrix} B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \end{pmatrix}$$

Therefore, for every pre-whitening key in $\mathcal{K}_{\text{wk}}$, the multiset $\mathcal{M}_{128}^{(3)}$ preserves its structure through pre-whitening and, after three rounds of Kalyna-128, all 16 output bytes are balanced. This proves the 3-round weak-key integral distinguisher for Kalyna-128.

Appendix D. Propagation of the 4-round integral distinguisher without pre-whitening for Kalyna-128

In this appendix, we present the propagation of the multiset $\mathcal{M}_{128}^{(4)}$ used in the 4-round integral distinguisher for Kalyna-128 when the pre-whitening part is omitted. Recall that $\mathcal{M}_{128}^{(4)}$ is obtained by propagating the multiset $\mathcal{M}_{128}^{(3)}$ backward through one more inverse round transformation. Although $\mathcal{M}_{128}^{(4)}$ does not admit a simple description in terms of the standard integral properties *all*, *balanced*, and *constant*, it is constructed so that after one forward round it evolves into the structured multiset

$$\mathcal{M}_{128}^{(3)} = \begin{pmatrix} A & C \\ A & C \\ A & C \\ A & C \\ C & A \\ C & A \\ C & A \\ C & A \end{pmatrix}$$

Since the pre-whitening key addition is omitted in this setting, the propagation starts directly from the first round.

Round 1.

Since $\mathcal{M}_{128}^{(4)}$ is obtained by applying one more inverse-round step to $\mathcal{M}_{128}^{(3)}$, we have

$$\text{SB}^{-1}(\text{SR}^{-1}(\text{MC}^{-1}(\mathcal{M}_{128}^{(3)}))).$$

Therefore, after one forward round, $\mathcal{M}_{128}^{(4)}$ evolves into a multiset having the same integral structure as $\mathcal{M}_{128}^{(3)}$. Equivalently, after the first round the multiset becomes

$$\mathcal{M}_{128}^{(3)} = \begin{pmatrix} A & C \\ A & C \\ A & C \\ A & C \\ C & A \\ C & A \\ C & A \\ C & A \end{pmatrix}$$

Round 2.

$$\begin{pmatrix} A & C \\ A & C \\ A & C \\ A & C \\ C & A \\ C & A \\ C & A \\ C & A \end{pmatrix} \xrightarrow{SB} \begin{pmatrix} A & C \\ A & C \\ A & C \\ A & C \\ C & A \\ C & A \\ C & A \\ C & A \end{pmatrix} \xrightarrow{SR} \begin{pmatrix} A & C \\ A & C \\ A & C \\ A & C \\ A & C \\ A & C \\ A & C \\ A & C \end{pmatrix} \xrightarrow{MC} \begin{pmatrix} A & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \end{pmatrix} \xrightarrow{\oplus K_2} \begin{pmatrix} A & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \end{pmatrix}$$

Round 3.

$$\begin{pmatrix} A & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \end{pmatrix} \xrightarrow{SB} \begin{pmatrix} A & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \end{pmatrix} \xrightarrow{SR} \begin{pmatrix} A & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \\ C & C \end{pmatrix} \xrightarrow{MC} \begin{pmatrix} A & C \\ A & C \\ A & C \\ A & C \\ A & C \\ A & C \\ A & C \\ A & C \end{pmatrix} \xrightarrow{\oplus K_3} \begin{pmatrix} A & C \\ A & C \\ A & C \\ A & C \\ A & C \\ A & C \\ A & C \\ A & C \end{pmatrix}$$

Round 4.

$$\begin{pmatrix} A & C \\ A & C \\ A & C \\ A & C \\ A & C \\ A & C \\ A & C \\ A & C \end{pmatrix} \xrightarrow{SB} \begin{pmatrix} A & C \\ A & C \\ A & C \\ A & C \\ A & C \\ A & C \\ A & C \\ A & C \end{pmatrix} \xrightarrow{SR} \begin{pmatrix} A & C \\ A & C \\ A & C \\ A & C \\ C & A \\ C & A \\ C & A \\ C & A \end{pmatrix} \xrightarrow{MC} \begin{pmatrix} B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \end{pmatrix} \xrightarrow{\oplus K_4} \begin{pmatrix} B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \\ B & B \end{pmatrix}$$

Therefore, when the pre-whitening part is omitted, all 16 output bytes are balanced after 4 rounds, which gives the 4-round integral distinguisher for Kalyna-128.

Appendix E. Propagation of the 2-round integral distinguisher for Kalyna-256

In this appendix, we present the detailed propagation of the multiset used in the 2-round integral distinguisher for Kalyna-256. Since the four byte positions $7, 15, 23$, and 31 are the most significant bytes of the four 64-bit columns, the corresponding cases are equivalent up to column symmetry. Hence, it is sufficient to describe the representative case where the active byte is chosen at position 7. We start from the multiset

$$\mathcal{M}_{256}^{(2)} = \begin{pmatrix} C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ A & C & C & C \end{pmatrix}.$$

Since the active byte is placed at a most significant byte position of a 64-bit column, Lemma 1 implies that the pre-whitening addition modulo 2^{64} preserves the above A/C structure.

Pre-whitening.

$$\mathcal{M}_{256} \xrightarrow{\boxplus K^{\mathrm{pw}}} \mathcal{M}_{256} \qquad \text{in terms of integral structure.}$$

Round 1.

$$\begin{pmatrix} C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ A & C & C & C \end{pmatrix} \xrightarrow{SB} \begin{pmatrix} C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ A & C & C & C \end{pmatrix} \xrightarrow{SR} \begin{pmatrix} C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & A \end{pmatrix} \xrightarrow{MC} \begin{pmatrix} C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \end{pmatrix} \xrightarrow{\oplus K_1} \begin{pmatrix} C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \end{pmatrix}$$

Round 2.

$$\begin{pmatrix} C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \end{pmatrix} \xrightarrow{SB} \begin{pmatrix} C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \end{pmatrix} \xrightarrow{SR} \begin{pmatrix} C & C & C & A \\ C & C & C & A \\ A & C & C & C \\ A & C & C & C \\ C & A & C & C \\ C & A & C & C \\ C & C & A & C \\ C & C & A & C \end{pmatrix} \xrightarrow{MC} \begin{pmatrix} B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \end{pmatrix} \xrightarrow{\oplus K_2} \begin{pmatrix} B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \end{pmatrix}$$

Therefore, the multiset $\mathcal{M}_{256}^{(2)}$ preserves its structure through pre-whitening and, after two rounds of Kalyna-256, all 32 output bytes are balanced. This proves the 2-round integral distinguisher for Kalyna-256.

Appendix F. Propagation of the 3-round weak-key integral distinguisher for Kalyna-256

In this appendix, we present the detailed propagation of the multiset used in the 3-round weak-key integral distinguisher for Kalyna-256. Since the possible choices of the two active bytes are equivalent up to symmetry, it is sufficient to consider the representative case in which the active bytes are chosen at positions 7 and 24. We start from the multiset

$$\mathcal{M}_{256}^{(3)} = \begin{pmatrix} C & C & C & A \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ A & C & C & C \end{pmatrix}.$$

We include the pre-whitening layer and consider the weak-key space

$$\mathcal{K}_{\mathrm{wk}} = \Big\{ K^{\mathrm{pw}} = (k_0, \ldots, k_{31}) \in (\mathbb{F}_2^8)^{32} \;\Big|\; k_{24} = 0 \Big\}.$$

Under this condition, the pre-whitening addition modulo 2^{64} preserves the bytewise A/C structure of $\mathcal{M}_{256}^{(3)}$. Indeed, the active byte at position 7 lies at a most significant byte position of its 64-bit column, so the modular addition does not destroy its active property. For the active byte at position 24, the condition $k_{24} = 0$ prevents a carry from being generated into byte 25, and hence bytes $25, \ldots, 31$ remain constant.

Pre-whitening.

$\mathcal{M}_{256}^{(3)} \xrightarrow{\boxplus K^{\mathrm{pw}}} \mathcal{M}_{256}^{(3)}$ for all $K^{\mathrm{pw}} \in \mathcal{K}_{\mathrm{wk}}$, in terms of integral structure.

Round 1.

$$\begin{pmatrix} C & C & C & A \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ A & C & C & C \end{pmatrix} \xrightarrow{SB} \begin{pmatrix} C & C & C & A \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ A & C & C & C \end{pmatrix} \xrightarrow{SR} \begin{pmatrix} C & C & C & A \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & A \end{pmatrix} \xrightarrow{MC} \begin{pmatrix} C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \end{pmatrix} \xrightarrow{\oplus K_1} \begin{pmatrix} C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \end{pmatrix}$$

Round 2.

$$\begin{pmatrix} C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \end{pmatrix} \xrightarrow{SB} \begin{pmatrix} C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \end{pmatrix} \xrightarrow{SR} \begin{pmatrix} C & C & C & A \\ C & C & C & A \\ A & C & C & C \\ A & C & C & C \\ C & A & C & C \\ C & A & C & C \\ C & C & A & C \\ C & C & A & C \end{pmatrix} \xrightarrow{MC} \begin{pmatrix} A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \end{pmatrix} \xrightarrow{\oplus K_2} \begin{pmatrix} A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \end{pmatrix}$$

Round 3.

$$\begin{pmatrix} A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \end{pmatrix} \xrightarrow{SB} \begin{pmatrix} A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \end{pmatrix} \xrightarrow{SR} \begin{pmatrix} A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \end{pmatrix} \xrightarrow{MC} \begin{pmatrix} B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \end{pmatrix} \xrightarrow{\oplus K_3} \begin{pmatrix} B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \end{pmatrix}$$

Therefore, for every pre-whitening key in $\mathcal{K}_{\text{wk}}$, the multiset $\mathcal{M}_{256}^{(3)}$ preserves its structure through pre-whitening and, after three rounds of Kalyna-256, all 32 output bytes are balanced. This proves the 3-round weak-key integral distinguisher.

Appendix G. Propagation of the 4-round weak-key integral distinguisher for Kalyna-256

In this appendix, we present the detailed propagation of the multiset used in the 4-round weak-key integral distinguisher for Kalyna-256. We start from the multiset

$$\mathcal{M}_{256}^{(4)} = \begin{pmatrix} A & C & C & A \\ A & C & C & A \\ C & C & A & A \\ C & C & A & A \\ C & A & A & C \\ C & A & A & C \\ A & A & C & C \\ A & A & C & C \end{pmatrix}.$$

By construction, this multiset is obtained from the 3-round input multiset $\mathcal{M}_{256}^{(3)}$ by one inverse-round propagation through

$$\text{SB}^{-1} \circ \text{SR}^{-1} \circ \text{MC}^{-1}.$$

We include the pre-whitening layer and consider the weak-key space

$$\mathcal{K}_{\text{wk}} = \left\{ K^{\text{pw}} = (k_0, \ldots, k_{31}) \in (\mathbb{F}_2^8)^{32} \;\middle|\; \begin{array}{l} k_0 = k_1 = k_6 = k_7 = k_{12} = k_{13} = k_{14} = k_{15} = 0, \\ k_{18} = k_{19} = k_{20} = k_{21} = k_{24} = k_{25} = k_{26} = k_{27} = 0 \end{array} \right\}.$$

Moreover, the fixed constants in the plaintext structure are chosen such that $c_2 = c_3 = c_4 = c_5 = c_8 = c_9 = c_{10} = c_{11} = c_{16} = c_{17} = 0$. Under these conditions, the pre-whitening addition modulo 2^{64} preserves the bytewise A/C structure of $\mathcal{M}_{256}^{(4)}$.

Pre-whitening.

$\mathcal{M}_{256}^{(4)} \xrightarrow{\boxplus K^{\text{pw}}} \mathcal{M}_{256}^{(4)}$ for all $K^{\text{pw}} \in \mathcal{K}_{\text{wk}}$, in terms of integral structure.

Round 1.

$$\mathcal{M}_{256}^{(4)} = \left(\text{SB}^{-1} \circ \text{SR}^{-1} \circ \text{MC}^{-1}\right)\left(\mathcal{M}_{256}^{(3)}\right),$$

Equivalently, after the first round the state has the same integral structure as

$$\mathcal{M}_{256}^{(3)} = \begin{pmatrix} C & C & C & A \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ A & C & C & C \end{pmatrix}.$$

Round 2.

$$\begin{pmatrix} C & C & C & A \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ A & C & C & C \end{pmatrix} \xrightarrow{SB} \begin{pmatrix} C & C & C & A \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ A & C & C & C \end{pmatrix} \xrightarrow{SR} \begin{pmatrix} C & C & C & A \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & A \end{pmatrix} \xrightarrow{MC} \begin{pmatrix} C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \end{pmatrix} \xrightarrow{\oplus K_2} \begin{pmatrix} C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \end{pmatrix}$$

Round 3.

$$\begin{pmatrix} C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \end{pmatrix} \xrightarrow{SB} \begin{pmatrix} C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \\ C & C & C & A \end{pmatrix} \xrightarrow{SR} \begin{pmatrix} C & C & C & A \\ C & C & C & A \\ A & C & C & C \\ A & C & C & C \\ C & A & C & C \\ C & A & C & C \\ C & C & A & C \\ C & C & A & C \end{pmatrix} \xrightarrow{MC} \begin{pmatrix} A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \end{pmatrix} \xrightarrow{\oplus K_3} \begin{pmatrix} A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \end{pmatrix}$$

Round 4.

$$\begin{pmatrix} A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \end{pmatrix} \xrightarrow{SB} \begin{pmatrix} A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \end{pmatrix} \xrightarrow{SR} \begin{pmatrix} A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \end{pmatrix} \xrightarrow{MC} \begin{pmatrix} B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \end{pmatrix} \xrightarrow{\oplus K_4} \begin{pmatrix} B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \end{pmatrix}$$

Therefore, for every pre-whitening key in $\mathcal{K}_{\text{wk}}$, the multiset $\mathcal{M}_{256}^{(4)}$ preserves its structure through pre-whitening and, after four rounds of Kalyna-256, all 32 output bytes are balanced. This proves the 4-round weak-key integral distinguisher.

Appendix H. Propagation of the 3-round integral distinguisher for Kalyna-512

In this appendix, we present the detailed propagation of the multiset used in the 3-round integral distinguisher for Kalyna-512. Since the byte positions $7, 15, 23, 31, 39, 47, 55$, and 63 are the most significant bytes of the eight 64-bit columns, the corresponding cases are equivalent up to column symmetry. Hence, it is sufficient to describe the representative case where the active byte is chosen at position 7. We start from the multiset

$$\mathcal{M}_{512}^{(3)} = \begin{pmatrix} C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ A & C & C & C & C & C & C & C \end{pmatrix}.$$

By Lemma 1, the pre-whitening operation preserves the above A/C structure.

Pre-whitening and Round 1.

$$\mathcal{M}_{512} \xrightarrow{\boxplus K^{pw}} \begin{pmatrix} C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ A & C & C & C & C & C & C & C \end{pmatrix} \xrightarrow{SB} \begin{pmatrix} C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ A & C & C & C & C & C & C & C \end{pmatrix} \xrightarrow{SR} \begin{pmatrix} C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & A \end{pmatrix} \xrightarrow{MC} \begin{pmatrix} C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \end{pmatrix} \xrightarrow{\oplus K_1} \begin{pmatrix} C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \end{pmatrix}$$

In the first round, the substitution layer preserves the active byte, since the S-boxes are bijective. For Kalyna-512, the ShiftRows operation shifts row i cyclically to the right by i positions. Hence, the active byte in row 7 is moved from the first column to the eighth column. The linear layer then diffuses this activity to all eight bytes of that column, while the round-key XOR preserves the integral type of each byte.

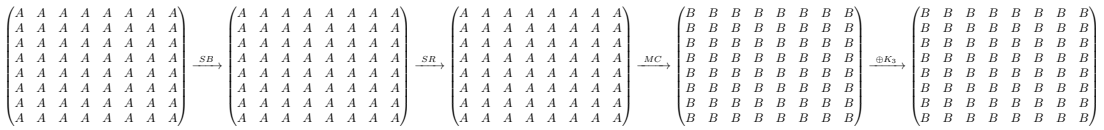
Round 2.

$$\begin{pmatrix} C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \end{pmatrix} \xrightarrow{SB} \begin{pmatrix} C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \end{pmatrix} \xrightarrow{SR} \begin{pmatrix} C & C & C & C & C & C & C & A \\ A & C & C & C & C & C & C & C \\ C & A & C & C & C & C & C & C \\ C & C & A & C & C & C & C & C \\ C & C & C & A & C & C & C & C \\ C & C & C & C & A & C & C & C \\ C & C & C & C & C & A & C & C \\ C & C & C & C & C & C & A & C \end{pmatrix} \xrightarrow{MC} \begin{pmatrix} A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \end{pmatrix} \xrightarrow{\oplus K_2} \begin{pmatrix} A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \end{pmatrix}$$

In the second round, the substitution layer again preserves the integral structure. After the ShiftRows operation, the eight active bytes are redistributed into eight distinct columns,

so each column contains exactly one active byte. The subsequent MixColumn transformation then diffuses the activity to all bytes of every column. Hence, after the second round, all 64 bytes satisfy the *All* property.

Round 3.



In the third round, the substitution layer and the ShiftRows operation preserve the *all* property of each byte. Since every *All* byte is also balanced, the linearity of the MixColumn transformation implies that every output byte after the linear layer is balanced. The final round-key XOR preserves this balanced structure. Therefore, after three rounds of Kalyna-512, all 64 output bytes are balanced, which proves the 3-round integral distinguisher.

Appendix I. Propagation of the 4-round weak-key integral distinguisher for Kalyna-512

In this appendix, we present the detailed propagation of the multiset used in the 4-round weak-key integral distinguisher for Kalyna-512. We start from the multiset

$$\mathcal{M}_{512}^{(4)} = \begin{pmatrix} A & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & A & C \\ C & C & C & C & C & A & C & C \\ C & C & C & C & A & C & C & C \\ C & C & C & A & C & C & C & C \\ C & C & A & C & C & C & C & C \\ C & A & C & C & C & C & C & C \end{pmatrix}.$$

As explained in the main text, this multiset is obtained from the 3-round input multiset by one inverse-round propagation. We include the pre-whitening layer and consider the weak-key space $\mathcal{K}_{\mathrm{wk}} = \left\{ K^{\mathrm{pw}} = (k_0, \ldots, k_{63}) \in (\mathbb{F}_2^8)^{64} \;\middle|\; k_0 = k_{15} = k_{22} = k_{29} = k_{36} = k_{43} = k_{50} = k_{57} = 0 \right\}$. Moreover, the fixed constants in the plaintext structure are chosen such that

$$c_{16} = c_{17} = c_{18} = c_{19} = c_{20} = c_{21} = 0,$$
$$c_{24} = c_{25} = c_{26} = c_{27} = c_{28} = 0,$$
$$c_{32} = c_{33} = c_{34} = c_{35} = 0,$$
$$c_{40} = c_{41} = c_{42} = 0, \qquad c_{48} = c_{49} = 0, \qquad c_{56} = 0.$$

Under these conditions, the pre-whitening addition modulo 2^{64} preserves the bytewise A/C structure of $\mathcal{M}_{512}^{(4)}$. Since 8 key bytes are fixed, we have

$$|\mathcal{K}_{\mathrm{wk}}| = 2^{448}.$$

Pre-whitening.

$\mathcal{M}_{512}^{(4)} \xrightarrow{\boxplus K^{\mathrm{pw}}} \mathcal{M}_{512}^{(4)}$ for all $K^{\mathrm{pw}} \in \mathcal{K}_{\mathrm{wk}}$, in terms of integral structure.

Round 1.

By construction, the multiset $\mathcal{M}_{512}^{(4)}$ is chosen so that, after one forward round, it evolves into the multiset used in the 3-round distinguisher. More precisely,

$$\mathcal{M}_{512}^{(4)} = \left(\mathrm{SB}^{-1} \circ \mathrm{SR}^{-1} \circ \mathrm{MC}^{-1}\right)\left(\mathcal{M}_{512}^{(3)}\right).$$

where

$$\mathcal{M}_{512}^{(3)} = \begin{pmatrix} C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ A & C & C & C & C & C & C & C \end{pmatrix}.$$

Thus, after the first round, the state has the same integral structure as the input multiset of the 3-round integral distinguisher.

Round 2.

$$\begin{pmatrix} C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ A & C & C & C & C & C & C & C \end{pmatrix} \xrightarrow{SB} \begin{pmatrix} C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ A & C & C & C & C & C & C & C \end{pmatrix} \xrightarrow{SR} \begin{pmatrix} C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & A \end{pmatrix} \xrightarrow{MC} \begin{pmatrix} C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \end{pmatrix} \xrightarrow{\oplus K_2} \begin{pmatrix} C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \end{pmatrix}$$

Round 3.

$$\begin{pmatrix} C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \end{pmatrix} \xrightarrow{SB} \begin{pmatrix} C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \end{pmatrix} \xrightarrow{SR} \begin{pmatrix} C & C & C & C & C & C & C & A \\ A & C & C & C & C & C & C & C \\ C & A & C & C & C & C & C & C \\ C & C & A & C & C & C & C & C \\ C & C & C & A & C & C & C & C \\ C & C & C & C & A & C & C & C \\ C & C & C & C & C & A & C & C \\ C & C & C & C & C & C & A & C \end{pmatrix} \xrightarrow{MC} \begin{pmatrix} A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \end{pmatrix} \xrightarrow{\oplus K_3} \begin{pmatrix} A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \end{pmatrix}$$

Round 4.

$$\begin{pmatrix} A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \end{pmatrix} \xrightarrow{SB} \begin{pmatrix} A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \end{pmatrix} \xrightarrow{SR} \begin{pmatrix} A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \end{pmatrix} \xrightarrow{MC} \begin{pmatrix} B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \end{pmatrix} \xrightarrow{\oplus K_4} \begin{pmatrix} B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \end{pmatrix}$$

Therefore, for every pre-whitening key in $\mathcal{K}_{\mathrm{wk}}$, the multiset $\mathcal{M}_{512}^{(4)}$ preserves its structure through pre-whitening and, after four rounds of Kalyna-512, all 64 output bytes are balanced. This proves the 4-round weak-key integral distinguisher.

Appendix J. Propagation of the 5-round integral distinguisher for Kalyna-512 without pre-whitening

In this appendix, we present the detailed propagation of the multiset used in the 5-round integral distinguisher for Kalyna-512 when the pre-whitening part is omitted.

Recall that the input multiset is obtained by one more inverse-round propagation from the 4-round input multiset

$$\mathcal{M}_{512}^{(4)} = \begin{pmatrix} A & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & A & C \\ C & C & C & C & C & A & C & C \\ C & C & C & C & A & C & C & C \\ C & C & C & A & C & C & C & C \\ C & C & A & C & C & C & C & C \\ C & A & C & C & C & C & C & C \end{pmatrix}.$$

More precisely,

$$\mathcal{M}_{512}^{(5)} = \left(\mathrm{SB}^{-1} \circ \mathrm{SR}^{-1} \circ \mathrm{MC}^{-1}\right)\left(\mathcal{M}_{512}^{(4)}\right).$$

At the level of byte-wise integral properties, this multiset can be represented as

$$\mathcal{M}_{512}^{(5)} = \begin{pmatrix} A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \end{pmatrix}.$$

However, this notation does not capture the dependencies among the bytes; the exact multiset is defined by the above inverse propagation. Since the pre-whitening operation is omitted in this setting, the propagation starts directly from the first round.

Round 1.

$\mathcal{M}_{512}^{(5)} \xrightarrow{\mathrm{SB}} \mathrm{SR}^{-1}\left(\mathrm{MC}^{-1}(\mathcal{M}_{512}^{(4)})\right) \xrightarrow{\mathrm{SR}} \mathrm{MC}^{-1}(\mathcal{M}_{512}^{(4)}) \xrightarrow{\mathrm{MC}} \mathcal{M}_{512}^{(4)} \xrightarrow{\oplus K_1} \mathcal{M}_{512}^{(4)}$ in terms of integral structure

Thus, after the first round the state has the same integral structure as

$$\mathcal{M}_{512}^{(4)} = \begin{pmatrix} A & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & A & C \\ C & C & C & C & C & A & C & C \\ C & C & C & C & A & C & C & C \\ C & C & C & A & C & C & C & C \\ C & C & A & C & C & C & C & C \\ C & A & C & C & C & C & C & C \end{pmatrix}.$$

Round 2.

$\mathcal{M}_{512}^{(4)} \xrightarrow{\mathrm{SB}} \mathrm{SR}^{-1}\left(\mathrm{MC}^{-1}(\mathcal{M}_{512}^{(3)})\right) \xrightarrow{\mathrm{SR}} \mathrm{MC}^{-1}(\mathcal{M}_{512}^{(3)}) \xrightarrow{\mathrm{MC}} \mathcal{M}_{512}^{(3)} \xrightarrow{\oplus K_2} \mathcal{M}_{512}^{(3)}$ in terms of integral structure

Equivalently, after the second round the state has the same integral structure as

$$\mathcal{M}_{512}^{(3)} = \begin{pmatrix} C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ A & C & C & C & C & C & C & C \end{pmatrix}.$$

Round 3.

$$\begin{pmatrix} C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ A & C & C & C & C & C & C & C \end{pmatrix} \xrightarrow{SB} \begin{pmatrix} C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ A & C & C & C & C & C & C & C \end{pmatrix} \xrightarrow{SR} \begin{pmatrix} C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & A \end{pmatrix} \xrightarrow{MC} \begin{pmatrix} C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \end{pmatrix} \xrightarrow{\oplus K_3} \begin{pmatrix} C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \end{pmatrix}$$

Round 4.

$$\begin{pmatrix} C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \end{pmatrix} \xrightarrow{SB} \begin{pmatrix} C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \\ C & C & C & C & C & C & C & A \end{pmatrix} \xrightarrow{SR} \begin{pmatrix} C & C & C & C & C & C & C & A \\ A & C & C & C & C & C & C & C \\ C & A & C & C & C & C & C & C \\ C & C & A & C & C & C & C & C \\ C & C & C & A & C & C & C & C \\ C & C & C & C & A & C & C & C \\ C & C & C & C & C & A & C & C \\ C & C & C & C & C & C & A & C \end{pmatrix} \xrightarrow{MC} \begin{pmatrix} A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \end{pmatrix} \xrightarrow{\oplus K_4} \begin{pmatrix} A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \end{pmatrix}$$

Round 5.

$$\begin{pmatrix} A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \end{pmatrix} \xrightarrow{SB} \begin{pmatrix} A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \end{pmatrix} \xrightarrow{SR} \begin{pmatrix} A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \end{pmatrix} \xrightarrow{MC} \begin{pmatrix} B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \end{pmatrix} \xrightarrow{\oplus K_5} \begin{pmatrix} B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B \end{pmatrix}$$

Therefore, when the pre-whitening part is omitted, all 64 output bytes are balanced after five rounds of Kalyna-512, which proves the 5-round integral distinguisher.