UDC 003.26:512.5

Refined Method of Impossible Differentials Search with Application to Kalyna-Like Ciphers

Andrii Turchyn[1,a], Serhii Yakovliev[1,b]

[1]*National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Institute of Physics and Technology*

Abstract

This work addresses the problem of evaluating the security of block ciphers against impossible differential cryptanalysis, with a particular focus on Kalyna-like ciphers. Based on formalized methods —specifically the Wu-Wang method — this work introduces refined rules tailored to AES- and Kalyna-like ciphers. These refinements simplify compatibility checks by replacing large systems of linear equations with computationally efficient conditions. Experimental results have identified several classes of impossible differentials for three-round versions of certain Kalyna cipher variants, thereby demonstrating the security of full-round ciphers against this method of cryptanalysis.

Keywords: symmetric cryptography, block ciphers, impossible differential cryptanalysis, Wu-Wang method, AES, Kalyna cipher

Introduction

Impossible differential cryptanalysis is a powerful method of evaluating the security of block ciphers. It uses differentials with zero probability to filter out incorrect keys. Since its introduction by Knudsen in 1998 [1] and many subsequent works (see, e.g., [2, 3, 4]), this approach has proven effective against many ciphers, including AES [5], Skipjack [4] and CLEFIA [6]. However, the method's success heavily depends on discovering long and complex impossible differentials, which remains challenging for word-oriented block ciphers such as Kalyna [7].

Formalized tools such as the $\mathcal{U}$-method [8] and the UID-method [9] have been widely used for finding impossible differentials. These tools rely on the miss-in-the-middle approach to identify inconsistencies in differential propagation. While effective for many ciphers, these methods often fail to detect longer impossible differentials or to handle the unique structures of ciphers like Kalyna, which feature independent S-boxes and MDS transformations.

Recent advances have sought to overcome these limitations. The $\mathcal{U}^*$-method proposed by Sun et al. [10] integrates constraint programming into the $\mathcal{U}$-method to exhaustively explore plaintext and ciphertext difference patterns. However, it does not account for contradictions in the S-box differential distribution tables (DDTs). Separately, Sun et al. [11] introduced the *primitive index* method, which estimates upper bounds on the length of impossible differentials in SPN-based ciphers by analyzing the structure of the linear layer while omitting S-box behaviour. Meanwhile, MILP-based approaches [12] have been adapted to model impossible differentials by embedding constraints on input-output difference pairs. While these approaches can detect general contradictions, they require the testing of a prohibitively large number of difference pairs.

A significant advancement was presented in [13], where a partitioning strategy was proposed for the MILP search space. This approach makes exhaustive analysis feasible. This method guarantees that if no impossible differentials are found within a given number of rounds, none exist — thus offering a form of provable security.

In this paper, we refine the Wu-Wang method [14] to address its limitations, simplifying compatibility checks and reducing computational complexity. We describe an improved practical procedure to automating the search of

[a]turchyn.andrew@gmail.com
[b]yasv@rl.kiev.ua

impossible differentials for Kalyna-like ciphers. Our experimental results demonstrate the effectiveness of this approach with Kalyna-128 and Kalyna-256 ciphers.

This paper is organized as follows. Section 1 introduces the formal notation of AES- and Kalyna-like SP-networks and the preliminaries required for impossible differential analysis. Section 2 describes the original Wu–Wang method and outlines its miss-in-the-middle compatibility checks. Section 3 refines the Wu–Wang framework for AES- and Kalyna-like SP-networks by introducing new B-rules that exploit the properties of MDS transformations used in AES-like ciphers. The practical procedure for the automated search of impossible differentials using the refined rules is then detailed. Section 4 presents the experimental results for some Kalyna cipher variants, highlighting the number and structure of the impossible differentials found.

1. Preliminaries

1.1. SP-Networks and Kalyna-like Ciphers

Denote by $V_u = \{0,1\}^u$ the space of binary vectors of length u; we name these vectors as "words". For $n = mu$ we identify the space V_n as $(V_u)^m$, and n-bit vectors as m-tuples of words.

An *n-bit substitution-permutation network* (or *SP-network*) is an iterative block cipher scheme with n-bit blocks, each round of which consists of three layers:

- addition with round keys (usually with XOR operation);
- nonlinear layer S with so-called *S-boxes* — bijective mappings of form $V_u \to V_u$, which transform words to words;
- linear layer L — some bijective transformation $V_n \to V_n$, linear w.r.t. XOR.

In *word-oriented SP-networks* the linear layer is normally defined as matrix multiplication over the space V_u or a finite field $\mathbb{F}_{2^u}$, thus all encryption rounds become some transformations over words.

The AES cipher [5] introduced a specific form of the byte-oriented SP-network. In *AES-like SP-networks* every state block is considered as a word matrix with ℓ rows and c columns (so $m = \ell c$), and the linear layer is divided into two subroutines:

- ShiftRows: words in every row of state are permuted;
- MixColumns: every column of state is multiplied by a given $\ell \times \ell$-matrix over $\mathbb{F}_{2^u}$.

The standardized version of AES uses $u = 8$ (bytes) and the state matrix with $c = \ell = 4$; Rijndael cipher, the predecessor of AES, also allows $\ell = 6$ and $\ell = 8$.

MixColumns transformation of AES is built upon a so-called MDS-matrix, which is characterized as follows. A *branch number* of $\ell \times \ell$-matrix M is defined as

$$B(M) = \min_{x \neq 0} \{wt(x) + wt(M \cdot x)\},$$

where $wt(x)$ is the number of non-zero words in vector x. The *MDS-matrix* M has the highest possible value of the branch number: $B(M) = \ell + 1$. The MixColumns transformation of AES uses an MDS matrix 4×4 with the branch number $B = 5$.

In 2015 the block cipher *Kalyna* [7] was adopted as Ukraine's national encryption standard DSTU 7624:2014, replacing the ageing GOST 28147-89 and offering far better software performance on modern 64-bit platforms. Kalyna is an iterated SP-network inspired by Rijndael/AES, but enlarged and re-parametrised to meet domestic security goals.

Kalyna cipher allows different block length and key sizes, as shown below:

Block (bit)	Key (bit)	Variant	Rounds r
128	128	128/128	10
128	256	128/256	14
256	256	256/256	14
256	512	256/512	18
512	512	512/512	18

The main differences between AES and Kalyna, which are relevant to our research, are as follow.

- Kalyna is tuned to 64-bit architectures, so its state matrix always has $\ell = 8$ (unlike $\ell = 4$ in AES); therefore, states of Kalyna have sizes 8×2, 8×4, 8×8.

This difference is crucial, since ShiftRows transformation of Kalyna cannot distribute words from one column to different columns of state, which affects avalanche effects and creates additional dependencies between internal words.

- The first and the last round keys are added to the state modulo 2^{64}, while others are added with XOR operation.
- MixColumns transformation of Kalyna is built upon 8×8 MDS-matrix with the branch number $B = 9$.

For a detailed description of Kalyna cipher we refer to [7].

The properties of Kalyna cipher allow to determine *Kalyna-like ciphers* as AES-like cipher with $\ell \leq c$.

In this paper, we use the notation "Kalyna-n" for model Kalyna-like ciphers with n-bit block and XOR addition with round keys (which corresponds to internal encryption rounds of Kalyna cipher). Note that key size doesn't matter for considered cryptanalytic methods due to usual assumption of round key independence.

In addition to the standardized Kalyna-128 and Kalyna-256 with state matrices of sizes 8×2 and 8×4, respectively, we also consider the reduced cipher Kalyna-64 with a 4×2 state matrix. Its ShiftRows function is analogous to that of Kalyna-128, while its MixColumns function is based on the AES MixColumns transformation.

Note, that wider 8-byte columns and a higher branch number dramatically enlarge the search space for classical $\mathcal{U}$-/UID- or naive MILP-methods in application to Kalyna cipher.

1.2. Formal Approach to Impossible Differential Cryptanalysis

The $\mathcal{U}$-method [8] and the UID-method [9] introduced the following formalization of impossible differential cryptanalysis of word-oriented block ciphers.

Let $(x_1, x_2, \ldots, x_n) \xrightarrow{r} (y_1, y_2, \ldots, y_n)$ denote an r-round difference transition in an n-word block cipher E. This notation implies that starting from the input difference $(x_1, \ldots, x_n)$ before round 1, after r rounds the cipher output difference $(y_1, \ldots, y_n)$ is a possible pattern. If no pair of actual plaintexts (or intermediate states) can implement such a transition, we call this differential an *impossible r-round differential* and denote it as

$$(x_1, \ldots, x_n) \overset{r}{\nrightarrow} (y_1, \ldots, y_n).$$

Suppose $U = (u_1, \ldots, u_n)$ is an input difference vector (each u_i corresponds to the difference of i-th input word), and let $U^r = (u_1^r, \ldots, u_n^r)$ be the difference after r rounds. Each u_i^r can be one of several types of word differences, as described next.

Following the convention in many word-oriented ciphers, we allow four basic labels:

- *Zero difference* (0): the difference word is exactly zero.
- *Nonzero fixed difference* (l_i): the difference word is nonzero but takes a fixed, predetermined value.
- *Nonzero variable difference* (m_i): the difference word is nonzero but may take any nonzero value.
- *Unknown variable difference* (r_i): the difference word may have any value (zero or nonzero).

Two difference vectors

$$U = (U_1, U_2, \ldots, U_n),$$
$$V = (V_1, V_2, \ldots, V_n),$$

are called *inconsistent* if there exists a nonempty subset $I \subseteq \{1, 2, \ldots, n\}$ such that

$$\bigoplus_{i\in I} U_i \neq \bigoplus_{i\in I} V_i, \tag{1}$$

under *all* possible assignments of variable differences. Intuitively, some fixed vs. variable differences cannot logically coincide, making these XOR sums impossible to match.

For example, if $U = (l_1 \oplus m_1, 0)$ and $V = (l_1, 0)$, where l_1 is nonzero fixed and m_1 is nonzero variable, then $l_1 \oplus m_1$ can never equal l_1, so U and V are inconsistent. Similarly, if $U = (l_1, l_1 \oplus m_1)$ and $V = (m_2, m_2)$, we find $u_1 \oplus u_2 = m_1$ and $v_1 \oplus v_2 = 0$, which cannot be equal; thus U and V are inconsistent again.

In an n-word block cipher E, let an input difference U^0 transit through i encryption rounds to become U^i. Likewise, an output difference V^0 can be traced backward through j decryption rounds to V^j. If at some point U^i and V^j are found to be *inconsistent* according to (1), then the $(i + j)$-round differential

$$U^0 \overset{i+j}{\nrightarrow} V^0$$

is impossible because the forward and backward paths for those words cannot be reconciled.

In the model considered, each round's transformations of words are categorized as one of three general types:

- *Zero transformation* (0): if the input difference is 0, the output difference remains 0.
- *Identity transformation* (1): no matter what difference is input, the output is unchanged.
- *Nonlinear transformation* (F): represents S-boxes or mixing steps. Typical rules include:
  - if input is 0, output is 0;
  - if input is l_i (nonzero fixed) or m_i (nonzero variable), output becomes some new m_j.
  - otherwise, output is r_j (unknown variable).

A schematic illustration of how input differences map to output differences under these three transformations is shown in Table 1.

In practice, these definitions provide a formal approach to identifying impossible differentials: we label each round's input/output difference as one of 0, l_i, m_i, or r_i, and check for mismatch that indicate no valid plaintext/ciphertext pair could implement that difference pattern. If such a mismatch arises after a total of $i + j$ rounds (forward and backward), the corresponding difference transition is deemed *impossible*.

2. A General Description of the Wu–Wang Method

The Wu-Wang method [14] is designed for the *automated search* of impossible differentials in various types of block ciphers. The main idea of the method involves the following steps.

1) *Constructing a formal model* of the cipher, where the input, output, and intermediate states (differences) are represented by sets of variables and relations. Each variable corresponds to some difference word and can be zero, nonzero, or unknown.

2) *Expressing the round function* of the cipher as a system of equations, divided into:

- *linear equations* (covering XOR, branching, linear transformations etc.);
- *nonlinear equations* (covering S-boxes, possibly key-dependent).

3) *Enumerating different input-output difference templates* and checking for contradictions by combining *linear-consistency checks* and an *information leakage* evaluation. If a contradiction is found, the corresponding differential is classified as impossible.

This method thus enables one to determine whether a given pair $(\Delta P, \Delta C)$ of differences of two plaintexts and two ciphertexts cannot arise during encryption, irrespective of the key.

Formal model of cipher involves several types of relations, which connected with every component of the encryption rounds as it given below.

Linear Transformations. Common linear operations in block ciphers include:

1) *branching*: $\Delta X = \Delta Y = \Delta Z$, represented by equations
$$\Delta x_i \oplus \Delta y_i = 0, \quad \Delta x_i \oplus \Delta z_i = 0;$$
2) *XOR*: $\Delta X \oplus \Delta Y = \Delta Z$, defined by $\Delta x_i \oplus \Delta y_i \oplus \Delta z_i = 0$;
3) *linear mixing*: $\Delta Y^T = M \cdot \Delta X^T$, where M is a matrix specifying the linear layer of encryption round (or some of its internal transformation); each word Δy_i depends linearly on several Δx_j, yielding equations of the form
$$\Delta y_i \oplus \bigoplus_j (m_{i,j} \cdot \Delta x_j) = 0.$$

Such relationships are linear and can be processed by standard linear-algebraic methods (e.g., Gaussian-Jordan elimination). Combining them across multiple rounds forms a large system of equations whose consistency is crucial in the Wu-Wang method.

Nonlinear Transformations. The differential transition through an S-box is denoted as
$$\Delta x_i \to \Delta y_i,$$
meaning if Δx_i is the input difference and Δy_i is the output difference, their relationship is specified by a set of (possibly non-linear) constraints. In [14], these are called "nonlinear equations."

A procedure for finding impossible differentials can be described as a sequence of the following steps.

- *Assigning Input-Output Templates.* One first designates which words of input and output differences $\Delta P \to \Delta C$ are zero (0) or nonzero (l_i). For example, one might explore all patterns with exactly one nonzero word, or two nonzero word, etc.
- *Combining Equations into a Single System.* All linear equations (branching, XOR, linear mixing) and nonlinear ones (S-box) are merged into a formal description. For a multi-round cipher,

Table 1
Types of formal transformations in word-oriented block ciphers

Transformation	Input	Output	Notes
0	$x \in \{0, l_i, m_i, r_i\}$	0	Zero transformation
1	$x \in \{0, l_i, m_i, r_i\}$	x	Identity transformation
F	0	0	F-transformation
	l_i	m_j	
	m_i	m_j	
	unknown	r_j	

the system captures the sequential passage of states through each round.

• *Checking Linear Consistency.* Using linear algebra (such as row reduction to echelon form), the method checks whether the linear part of formal system has a solution. If no solution exists, the input-output difference $\Delta P \to \Delta C$ cannot occur, therefore this differential is classified as *impossible*.

Note that this step is the main innovation of the Wu-Wang method: linear constraints are explicitly tested for possible (in)consistency, unlike the earlier UID-method.

• *Miss-in-the-middle Contradictions.* If the linear system is consistent, the algorithm propagates these differences forward and backward through all stages of the cipher, applying S-box constraints and other rules (e.g., zero-in $\to$ zero-out). If at some point a variable is forced to be both zero and nonzero, we have a *contradiction*, implying an impossible differential.

• *Conclusion.* Once a contradiction is found (due to an unsolvable linear system or inconsistent assignments in intermediate variables), the differential is labeled "impossible." Conversely, if no contradiction arises, the method does not exclude the possibility that the given differential might be impossible.

Hence, the Wu-Wang method supports combining arbitrarily complex linear transformations (e.g., MDS) and S-box-based nonlinearities, checking whether certain long differential paths can occur. It does not necessarily increase the number of rounds for which impossible differentials are found but automates the detection of unreachable difference patterns.

3. Improved Formalization of the Wu-Wang Method for AES-like and Kalyna-like Ciphers

We describe three types of formal rules for the Wu-Wang method, applied to AES-like SP-networks. Two of them replicates rules from [14], and the last one is introduced in this work.

1) *Non-linear transformation (NL-rules):* For each input word Δx_i, a new variable Δy_i is introduced, representing the output difference after passing through the S-box. For each pair $(\Delta x_i, \Delta y_i)$, a non-linear rule is added:

$$\Delta x_i \to \Delta y_i.$$

For example, in Kalyna-128 each round adds 16 non-linear rules.

2) *Linear transformation (L-rules):* After applying ShiftRows to the state Δy, the resulting state is $\Delta y'$, and after applying MixColumns to the state $\Delta y'$, the resulting state is Δz. Each column $\Delta y'_j$ is multiplied by an MDS matrix M, producing the output vector Δz_j:

$$\Delta z_j = M \cdot \Delta y'_j.$$

For example, in Kalyna-128 linear transformation generates 16 linear equations per round.

3) *B-rules:* For each column j, the B-rule $B[\Delta y', \Delta z]$ checks the number of non-zero words among both input and output variables:

$$wt(\Delta y'_j) + wt(\Delta z_j) \geq B,$$

where B is the branch number of the MDS matrix M (this value is fixed for given cipher). For Kalyna-128, where the state matrix has two 8-byte columns, each round adds two B-rules, each with 16 corresponding variables and $B = 9$.

The proposed B-rules add valuable insights into the relationship between input and output variables in MDS transformations, streamlining

linear-system compatibility checks. The Wu-Wang method checks the compatibility of the linear system only at the beginning of the analysis. However, during the search for contradictions and the assignment of intermediate variables, changes in the system may occur, potentially leading to contradictions. Evaluating these conditions can provide additional insights into differential propagation and possible inconsistencies in the cipher. Moreover, the following properties of AES-like SP-networks with MDS-transformations allow significant simplification of the linear consistency checking step.

- *System decomposition:* Linear equations for each column on every round are independent, allowing the overall linear system to be split into several smaller subsystems for each round. This significantly reduces the complexity of the analysis.
- *Guarantee of solvability:* MDS matrices ensure that if the number of non-zero words at the input and output is at least B, the system has a solution. For example, Kalyna-128 uses 8×8 MDS matrix in MixColumns, therefore the system is solvable when number of non-zero variables at the input and output of every column is not less than 9.
- *Compatibility checks:* Instead of solving the entire MDS system, the process is reduced to counting the number of non-zero variables in the input and output vectors for each column, which is described as B-rule.

Therefore, the proposed approach with B-rules enables the Wu-Wang method to be applied more effectively to AES-like and Kalyna-like ciphers.

The method for finding impossible differentials in ciphers involves the use of formal rules that correspond to the structure of the cipher. The algorithm proceeds in several stages, sequentially processing various types of rules. The general idea is to establish values for input and output variables, verify compliance with defined rules, and identify contradictions in the differential path.

In general refined Wu-Wang method can be described as a sequence of the following steps.

• *Construction of Formal Cipher Model.* For a given number of rounds r a set of formal rules, which describes differential transitions in cipher E, is determined.

• *Initialization.* At the first stage, a set of templates for input and output variables is selected. These variables can take values "zero" (0) or "non-zero fixed" (l_i). To reduce computational complexity, it is recommended to limit templates to those with a small number of non-zero variables.

• *Processing NL-Rules.* At this stage, nonlinear equations associated with the cipher's S-boxes are processed. For each equation $x \to y$, the following checks are performed:

- If x is a zero variable and y is non-zero, or vice versa, a contradiction is found. The rule processing stops, and the differential is determined to be impossible.
- If x is a zero or non-zero variable and y is unknown, y is set to the value of x. Similarly, if y is a zero or non-zero variable and x is unknown, x is set to the value of y.

• *Processing L-Rules.* At this stage, linear equations related to MDS matrices and linear transformations are processed. For each equation, the following checks are performed:

- If the equation is of the form
$$a \cdot x = 0$$
and x is a non-zero variable, a contradiction is found. The rule processing stops, and the differential is determined to be impossible.
- If the equation is of the form
$$a \cdot x = \text{const} \neq 0$$
and x is a zero variable, a contradiction is found. The rule processing stops.
- If the equation is of the form
$$a \cdot x = \text{const}$$
and x is unknown, x is set to const (zero if const $= 0$, and non-zero if const $\neq 0$).
- If the equation is of the form
$$a \cdot x \oplus b \cdot y = 0,$$
and x is a non-zero variable while y is unknown, y is set to non-zero. Similarly, if y is a non-zero variable while x is unknown, x is set to non-zero.

• *Processing B-Rules.* For each B-rule $[y, z]$, the following checks are performed:

- If all variables in the vector y are zero, all variables in z are set to zero, and vice versa.

Table 2
An example of difference propagation and mismatch in 3-round cipher Kalyna-64.

Round		Input	
		0000	0001
1	SB:	0000	0001
	SR:	0001	0000
	MC:	1111	0000
2	SB:	1111	0000
	SR:	1100	0011
	MC:	2200	0022
3	SB:	2200	0022
	SR:	2222	0000
	MC:	0001	0000

Table 3
An example of difference propagation and mismatch in 3-round cipher Kalyna-128

Round		Input			
		0000	0000	0000	0101
1	SB:	0000	0000	0000	0101
	SR:	0000	0101	0000	0000
	MC:	2222	2222	0000	0000
2	SB:	2222	2222	0000	0000
	SR:	2222	0000	0000	2222
	MC:	2222	0000	0000	2222
3	SB:	2222	0000	0000	2222
	SR:	2222	2222	0000	0000
	MC:	0000	0001	0000	0000

- If the total number of zero variables in y and z exceeds $B-1$, a contradiction is found. The rule processing stops, and the differential is determined to be impossible.

• *Iterative Processing.* The processing of all types of rules is repeated iteratively until no new information about intermediate variables can be established. If, after a complete cycle of checks, no new values are determined, the algorithm terminates. If no contradiction is found, nothing can be concluded about the differential.

4. Experimental results

To illustrate the effectiveness of the proposed refinement to the Wu-Wang method we apply it to the Kalyna-n ciphers. At first we considered reduced Kalyna-64 cipher, then Kalyna-128 and Kalyna-256 ciphers due to a form of their state matrices with $\ell < c$.

Results for the Kalyna-64. For the three-round version of the Kalyna-64 cipher all possible templates for input and output difference words were analyzed. As a result, 900 classes of truncated impossible differentials were found. It was identified that, in all these differentials, contradictions arose during the processing of B-rules, with the contradiction being detected after just two iterations of rule processing. For differentials where the methodology could not determine whether they were impossible or not, the processing typically completed in three iterations (in rare cases, four iterations).

For the four-round version of Kalyna-64 cipher no impossible differentials were found.

Table 2 provides an example of the differential path for a three-round impossible differential. Here, `0` represents a zero variable, `1` represents a non-zero variable, and `2` represents an unknown variable. The input vector shows the input difference values, and the area where the contradiction was detected is highlighted.

Results for the Kalyna-128 and the Kalyna-256. For the Kalyna-128 and the Kalyna-256 ciphers, only templates with input and output vectors of weight no more than 2 were considered. For the three-round ciphers, the following results were obtained:

- *Kalyna-128*: 5,184 classes of impossible differentials out of 18,496 templates analyzed;
- *Kalyna-256*: 278,784 classes of impossible differentials, which constituted all analyzed templates.

Similar to the reduced 64-bit version, in all identified impossible differentials contradictions arose during the processing of B-rules and were detected after just two iterations. Interestingly, all the discovered impossible differentials included at least one fully zeroed column in the output. This observation suggests that for the Kalyna-256 cipher, there may exist classes of impossible differentials with differences of weight 3 in both input and output (guaranteeing zeroed columns), and potentially even with higher weights.

Table 4
An example of difference propagation and mismatch in 3-round cipher Kalyna-256

Round		Input							
		1000	0000	0000	0000	0000	0000	0000	0000
1	SB:	1000	0000	0000	0000	0000	0000	0000	0000
	SR:	1000	0000	0000	0000	0000	0000	0000	0000
	MC:	1111	1111	0000	0000	0000	0000	0000	0000
2	SB:	1111	1111	0000	0000	0000	0000	0000	0000
	SR:	1100	0000	0000	0011	0000	1100	0000	0011
	MC:	0022	0000	2200	0000	0000	0022	0000	2200
3	SB:	0022	0000	2200	0000	0000	0022	0000	2200
	SR:	0000	0000	2222	2222	0000	0000	0000	0000
	MC:	0000	0000	1000	0000	0000	0000	0000	0000

Examples of differential paths for the discovered impossible differentials are provided in Table 3 and Table 4.

The obtained results demonstrate that the full-round variants of the Kalyna ciphers (Kalyna-128 with 10 rounds and Kalyna-256 with 14 rounds) can be considered as secure against impossible differential cryptanalysis.

Conclusions

In this work, we considered the Wu-Wang method, one of the most powerful approaches for finding impossible differentials in word-oriented ciphers. It has been shown that, for AES- and Kalyna-like SP-networks, this method can be simplified and enhanced by performing additional checks on the properties of MDS transformations during the analysis of differential paths within the cipher.

The proposed refinement was used to discover classes of three-round impossible differentials for the Kalyna-128 and Kalyna-256 ciphers. In particular, it was demonstrated that all differentials with two non-zero input and output words are impossible in the three-round Kalyna-256 cipher.

Future research may focus on refining these methods to evaluate the probabilities of truncated differentials and finding impossible differentials for specific classes of block ciphers. This would require more detailed studies of the properties of their algebraic structures.

References

[1] L. R. Knudsen, “Deal — a 128-bit block cipher,” Tech. Rep. Technical Report 151, Department of Informatics, University of Bergen, 1998. https://www.ii.uib.no/~larsr/papers/Knudsen98.pdf.

[2] E. Biham and N. Keller, “Cryptanalysis of Reduced Variants of Rijndael,” 2002. http://madchat.fr/crypto/codebreakers/35-ebiham.pdf.

[3] E. Biham, A. Biryukov, and A. Shamir, “Miss in the middle attacks on idea, khufu and khafre,” in Fast Software Encryption (FSE’99), vol. 1636 of Lecture Notes in Computer Science, pp. 124–138, Springer, 1999.

[4] E. Biham, A. Biryukov, and A. Shamir, “Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials,” in Advances in Cryptology – EUROCRYPT’99, vol. 1592 of Lecture Notes in Computer Science, pp. 12–23, Springer, 1999.

[5] J. Daemen and V. Rijmen, The Design of Rijndael: AES – The Advanced Encryption Standard. Springer-Verlag, 2002.

[6] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, “The 128-bit blockcipher clefia (extended abstract),” in Fast Software Encryption (FSE 2007), vol. 4593 of Lecture Notes in Computer Science, pp. 181–195, Springer, 2007.

[7] R. Oliynykov, I. Gorbenko, O. Kazymyrov, et al., "A New Encryption Standard of Ukraine: The Kalyna Block Cipher," Cryptology ePrint Archive, 2015. http://eprint.iacr.org/2015/650.

[8] J. Kim, S. Hong, J. Sung, C. Lee, and S. Lee, "Impossible differential cryptanalysis for block cipher structures," in Progress in Cryptology – INDOCRYPT 2003 (T. Johansson and S. Maitra, eds.), vol. 2904 of Lecture Notes in Computer Science, pp. 82–96, Springer, 2003.

[9] Y. Luo, Z. Wu, X. Lai, and G. Gong, "A Unified Method for Finding Impossible Differentials of Block Cipher Structures," Cryptology ePrint Archive, 2009. http://eprint.iacr.org/2009/627.

[10] L. Sun, D. Gérault, W. Wang, and M. Wang, "On the Usage of Deterministic (Related-Key) Truncated Differentials and Multidimensional Linear Approximations for SPN Ciphers," IACR Transactions on Symmetric Cryptology, vol. 2020, no. 3, pp. 262–287, 2020.

[11] B. Sun, M. Liu, J. Guo, V. Rijmen, and R. Li, "Provable Security Evaluation of Structures Against Impossible Differential and Zero Correlation Linear Cryptanalysis," in Advances in Cryptology – EUROCRYPT 2016, Part I, vol. 9665 of Lecture Notes in Computer Science, pp. 196–213, Springer, 2016.

[12] Y. Sasaki and Y. Todo, "New Impossible Differential Search Tool from Design and Cryptanalysis Aspects: Revealing Structural Properties of Several Ciphers," in Advances in Cryptology – EUROCRYPT 2017, Part III, vol. 10212 of Lecture Notes in Computer Science, pp. 185–215, Springer, 2017.

[13] K. Hu, T. Peyrin, and M. Wang, "Finding All Impossible Differentials When Considering the DDT," Cryptology ePrint Archive, 2022. https://eprint.iacr.org/2022/1034.

[14] S. Wu and M. Wang, "Automatic Search of Truncated Impossible Differentials for Word-Oriented Block Ciphers," Cryptology ePrint Archive, 2012. http://eprint.iacr.org/2012/214.