# Crypto Term Paper

## Kalyna

Shivam Sharma, Himanshu Sekhar Nayak and Ayush Gupta

Indian Institute of Technology Bhilai, Raipur, India

Himanshu Sekhar Nayak himanshun@iitbhilai.ac.in
Shivam Sharma shivamsharma@iitbhilai.ac.in
Ayush Gupta ayushg@iitbhilai.ac.in

**Abstract.** Main requirements for Kalyna were both high security level and high performance of software implementation on general-purpose 64-bit CPUs.The cipher has SPN-based (Rijndael-like) structure with increased MDS matrix size, a new set of four different S-boxes, pre- and postwhitening using modulo $2^64$ addition and a new construction of the key schedule. Kalyna supports block size and key length of 128, 256 and 512 bits (key length can be either equal or double of the block size).

**Keywords:** Kalyna

## 1 Introduction

The Kalyna block cipher was selected during Ukrainian National Public Cryptographic Competition (2007-2010) and its slight modification was approved as the new encryption standard of Ukraine in 2015. Kalyna is a symmetric block cipher. It supports block sizes of 128, 256 or 512 bits; the key length is either equal to or double the block size. Kalyna has 10 rounds for 128-bit keys, 14 rounds for 256-bit keys and 18 rounds for 512-bit keys. The $8 \times c$ matrix is the cipher internal state. For 128 bit block size $c$ is 2 and for 256 bit block size $c$ is 4. For 256 bit and 512 bit state size, state matrix grows by adding extra 2 columns respectively. Each cell contains 1-Byte and the message is filled from top to bottom and column by column order.

| # | Block Size | Key Length | Rounds |
|---|---|---|---|
| 1 | 128 | 128 | 10 |
| 2 | 128 | 256 | 14 |
| 3 | 256 | 256 | 14 |
| 4 | 256 | 512 | 18 |
| 5 | 512 | 512 | 18 |

**Figure 1:** Number of Rounds

# 2 Symbols and notations

The following notations are used in the standard.

- $GF(2^8)$ - the finite field with the irreducible polynomial $x^8 + x^4 + x^3 + x^2 + 1$
- $F^{A1,A2,\cdots}$ (F is Function that takes argument $A_1, A_2, ..., A_n$)
- $F = F_1 \circ F_2 \circ F_3 ...$ (F is evaluated from right to left)
- $K_n$ - $n^{th}$ round key
- $\eta^{K_n}$ - Addition modulo $2^{64}$ with round key $K_n$ and the state matrix
- $\gamma$ - SubBytes
- $\pi$ - ShiftRows
- $\theta$ - MixColumns
- $\zeta^{K_n}$ - Modulo 2 addition of the round key $K_n$ and the state matrix
- $\eta_{-1}^{K_n}$ - Subtraction modulo $2^{64}$ with round key $K_n$ and the state matrix
- $\gamma_{-1}$ - Inverse S-Box
- $\pi_{-1}$ - Inverse ShiftRows
- $\theta_{-1}$ - Inverse MixColumns
- $G$ - State matrix
- $cell_{i \times j}$ -
- $g_{i,j}$ - $i^{th}$ row and $j^{th}$ column cell

# 3 Encryption

## 3.1 Structure of the basic encryption transformation

The basic encryption transformation is defined in the following way:

$$\eta^{K_r} \circ \theta \circ \pi \circ \gamma \circ \prod_{i=1}^{r-1} (\zeta^{K_i} \circ \theta \circ \pi \circ \gamma) \circ \eta^{K_0}$$

## 3.2 Function Of Addition Modulo $2^{64}$ ($\eta^{K_n}$)

This function operates on the columns of the state matrix and and the round key matrix. $i^{th}$ column of the state matrix is added with $i^{th}$ column of the round-key. The modulo $2^{64}$ of the result is the $i^{th}$ column of the output. For addition little-endian convention is used (i.e. byte of the top cell is the stored as least significant byte). The result is a matrix of size of $8 \times c$.

## 3.3 Layer Of Non-Linear Bijective Mapping ($\gamma$)

$\gamma$ implements the SubBytes operation on the state matrix. There are 4 S-Boxes used in Kalyna, these S-boxes follows a predefined order in the implementation of SubBytes operation. Each byte $g_{i,j}$ of the state matrix is substituted by S-Box based on value of row number $\mod 4$ ($i \mod 4$).

## 3.4 Permutation Of Elements In The State ($\pi$)

$\pi$ performs cyclic right shift for the rows of the state matrix. The value of shift depends on the row number $i$ and the block size $l \in 128, 256, 512$, and is calculated accordingly by expression $\lfloor \frac{i.l}{512} \rfloor$.

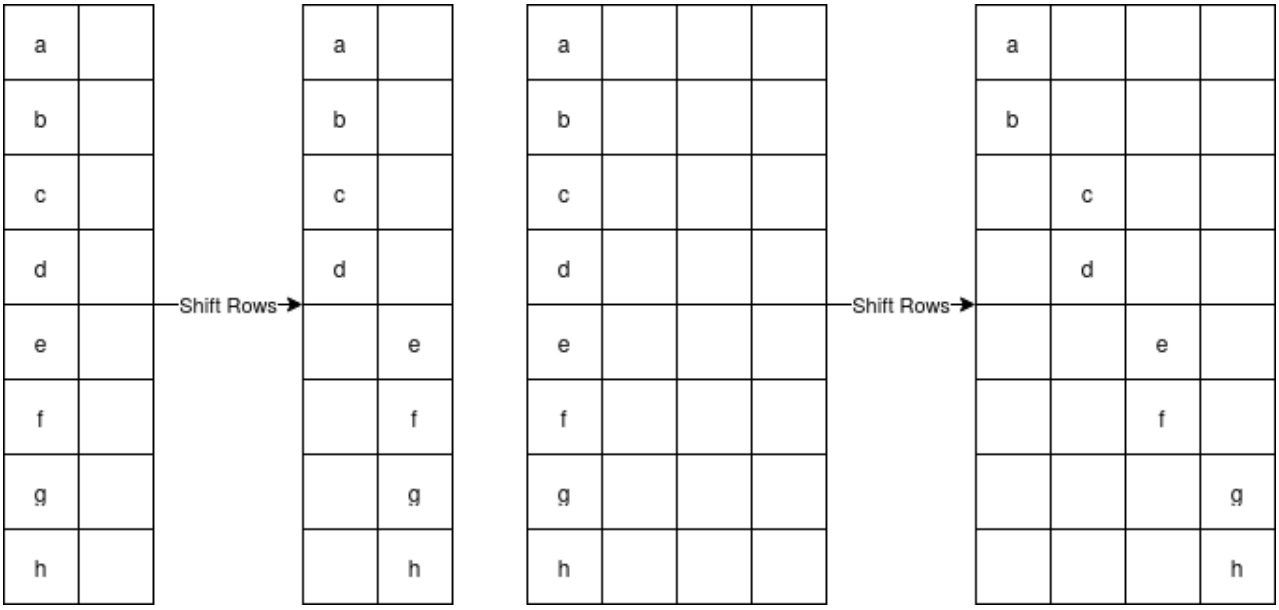


**Figure 2:** Shift Rows

## 3.5 Linear transformation ($\theta$)

This function $\theta$ calculates the each byte of the resulting state matrix $G' = (g'_{i,j})$ over $GF(2^8)$ according to the formula

$$g'_{i,j} = (M >>> i) \otimes G_j$$

, where $M = (0x01, 0x01, 0x05, 0x01, 0x08, 0x06, 0x07, 0x04)$ is the vector that forms the circulant matrix with the MDS property, and $G_j$ is the $j^{th}$ column of the state matrix G. The vector M consists of the hexadecimal constants (bytes) that are elements of the finite field $GF(2^8)$. The right circular shift is made with respect to elements of the set M.

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} = \begin{bmatrix} 01.a_0 \oplus 01.a_1 \oplus 05.a_2 \oplus 01.a_3 \oplus 08.a_4 \oplus 06.a_5 \oplus 07.a_6 \oplus 04.a_7 \\ 04.a_0 \oplus 01.a_1 \oplus 01.a_2 \oplus 05.a_3 \oplus 01.a_4 \oplus 08.a_5 \oplus 06.a_6 \oplus 07.a_7 \\ 07.a_0 \oplus 04.a_1 \oplus 01.a_2 \oplus 01.a_3 \oplus 05.a_4 \oplus 01.a_5 \oplus 08.a_6 \oplus 06.a_7 \\ 06.a_0 \oplus 07.a_1 \oplus 04.a_2 \oplus 01.a_3 \oplus 01.a_4 \oplus 05.a_5 \oplus 01.a_6 \oplus 08.a_7 \\ 08.a_0 \oplus 06.a_1 \oplus 07.a_2 \oplus 04.a_3 \oplus 01.a_4 \oplus 01.a_5 \oplus 05.a_6 \oplus 01.a_7 \\ 01.a_0 \oplus 08.a_1 \oplus 06.a_2 \oplus 07.a_3 \oplus 04.a_4 \oplus 01.a_5 \oplus 01.a_6 \oplus 05.a_7 \\ 05.a_0 \oplus 01.a_1 \oplus 08.a_2 \oplus 06.a_3 \oplus 07.a_4 \oplus 04.a_5 \oplus 01.a_6 \oplus 01.a_7 \\ 01.a_0 \oplus 05.a_1 \oplus 01.a_2 \oplus 08.a_3 \oplus 06.a_4 \oplus 07.a_5 \oplus 04.a_6 \oplus 01.a_7 \end{bmatrix}$$

## 3.6 Function of addition modulo 2 ($\zeta^{K_n}$)

This function $\zeta^{K_n}$ performs bit-wise XOR operation between the state matrix $G$ and the round key matrix $K_n$. The result is a matrix of $8 \times c$ size.

# 4 Decryption

## 4.1 Structure of the basic decryption transformation

The basic decryption transformation is defined in the following way:

$$\eta_{-1}^{K_0} \circ \prod_{i=r-1}^{1} (\gamma_{-1} \circ \pi_{-1} \circ \theta_{-1} \circ \zeta^{K_i}) \circ \gamma_{-1} \circ \pi_{-1} \circ \theta_{-1} \circ \eta_{-1}^{K_r}$$

## 4.2 Function Of Subtraction Modulo $2^{64}(\eta_{-1}^{K_n})$

This function operates on the columns of the state matrix and and the round key matrix. $i^{th}$ column of the state matrix is subtracted with $i^{th}$ column of the round-key. The modulo $2^{64}$ of the result is the $i^{th}$ column of the output. For subtraction little-endian convention is used (i.e. byte of the top cell is the stored as least significant byte). The result is a matrix of size of $8 \times c$.

## 4.3 Layer Of Inverse Non-Linear Bijective Mapping($\gamma_{-1}$)

$\gamma_{-1}$ implements the Inverse SubBytes operation on the state matrix. There are 4 Inverse S-Boxes used in Kalyna, these Inverse S-boxes follows a predefined order in the implementation of Inverse SubBytes operation. Each byte $g_{i,j}$ of the state matrix is substituted by Inverse S-Box based on value of row number $\mod 4$ ($i \mod 4$).

## 4.4 Inverse Permutation Of Elements($\pi_{-1}$)

$\pi_{-1}$ performs cyclic left shift for the rows of the state matrix. The value of shift depends on the row number $i$ and the block size $l \in 128, 256, 512$, and is calculated accordingly by expression $\lfloor \frac{i.l}{512} \rfloor$.

## 4.5 Inverse Linear Transformation($\theta_{-1}$)

This function $\theta_{-1}$ calculates the each byte of the resulting state matrix $G' = (g'_{i,j})$ over $GF(2^8)$ according to the formula

$$g'_{i,j} = (M^{-1} >>> i) \otimes G_j$$

, where $M^{-1} = (0xad, 0x95, 0x76, 0xa8, 0x2f, 0x49, 0xd7, 0xca)$ is the vector that forms the circulant matrix with the MDS property, and $G_j$ is the $j^{th}$ column of the state matrix G. The vector $M^{-1}$ consists of the hexadecimal constants (bytes) that are elements of the finite field $GF(2^8)$. The right circular shift is made with respect to elements of the set $M^{-1}$.

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} = \begin{bmatrix} ad.a_0 \oplus 95.a_1 \oplus 76.a_2 \oplus a8.a_3 \oplus 2f.a_4 \oplus 49.a_5 \oplus d7.a_6 \oplus ca.a_7 \\ ca.a_0 \oplus ad.a_1 \oplus 95.a_2 \oplus 76.a_3 \oplus a8.a_4 \oplus 2f.a_5 \oplus 49.a_6 \oplus d7.a_7 \\ d7.a_0 \oplus ca.a_1 \oplus ad.a_2 \oplus 95.a_3 \oplus 76.a_4 \oplus a8.a_5 \oplus 2f.a_6 \oplus 49.a_7 \\ 49.a_0 \oplus d7.a_1 \oplus ca.a_2 \oplus ad.a_3 \oplus 95.a_4 \oplus 76.a_5 \oplus a8.a_6 \oplus 2f.a_7 \\ 2f.a_0 \oplus 49.a_1 \oplus d7.a_2 \oplus ca.a_3 \oplus ad.a_4 \oplus 95.a_5 \oplus 76.a_6 \oplus a8.a_7 \\ a8.a_0 \oplus 2f.a_1 \oplus 49.a_2 \oplus d7.a_3 \oplus ca.a_4 \oplus ad.a_5 \oplus 95.a_6 \oplus 76.a_7 \\ 76.a_0 \oplus a8.a_1 \oplus 2f.a_2 \oplus 49.a_3 \oplus d7.a_4 \oplus ca.a_5 \oplus ad.a_6 \oplus 95.a_7 \\ 95.a_0 \oplus 76.a_1 \oplus a8.a_2 \oplus 2f.a_3 \oplus 49.a_4 \oplus d7.a_5 \oplus ca.a_6 \oplus ad.a_7 \end{bmatrix}$$

# 5 Round Key Generation

Kalyna Generates an Intermediate key($K_\sigma$) which is used to generates round keys. The length of $K_\sigma$ is same as the length of state and its representation is same as state matrix.

$$K_\sigma = \theta \circ \pi \circ \gamma \circ \eta^{(K_\alpha)} \circ \theta \circ \pi \circ \gamma \circ \zeta^{(K_\omega)} \circ \theta \circ \pi \circ \gamma \circ \eta^{(K_\alpha)}$$

If the state size and the key length is equal then $K_\alpha = K_\omega = K$. If the block size and key length are not equal (length of key size is double the state size) then $K_\alpha$ = left half of the key and $K_\omega$ = right half of the key.

Every $2i + 1$ round key is generated from $2i$ round key, where $i \in 0 \cup Z^+$.

## 5.1 Even round key generation

Even round key generation depends upon key $K$, intermediate round key $K_\sigma$, and the index i. The transformation function is represented as follows:

$$\Xi^{(K,K_\sigma,i)} = \eta^{\phi_i^{(K_\sigma)}} \circ \theta \circ \pi \circ \gamma \circ \zeta^{\phi_i^{(K_\sigma)}} \circ \theta \circ \pi \circ \gamma \circ \eta^{\phi_i^{(K_\sigma)}}$$
$$\phi_i^{(K_\sigma)} = \eta^{K_\sigma}(\nu << (i/2))$$
$$\nu = \mu^{0x00010001...0001}$$

$\nu$ is same length as length of the internal state. $\mu$ returns the little-endian version of the input hexadecimal number.

**if** block length equals key length **then**
  $K = K >>> 32.i$
**else**
  **if** $i$ is divisible by 4 **then**
    $K = K >>> 16.i$
  **else**
    $K = K >>> 64\lfloor i/4 \rfloor$
  **end if**
**end if**

## 5.2 Odd round key generation

$i^{th}$ round key is generated from $i - 1^{th}$ round keys according to the formula:

$$K_i = (K_{i-1} <<< 2N_b + 3)$$

$N_b$ is the number of bytes in the state and $i \in \{1, 3, 5, ..., t - 1\}$.

# 6 Cryptanalysis

## 6.1 Differential Cryptanalysis

Here are some statistics about four S-boxes of Kalyna.

| DDT Info for S-Box 1 | |
|---|---|
| Values | Number of values in the difference table |
| 8 | 15 |
| 6 | 246 |
| 4 | 3423 |
| 2 | 24996 |
| 0 | 36345 |

| DDT Info for S-Box 2 | |
|---|---|
| Values | Number of values in the difference table |
| 8 | 9 |
| 6 | 191 |
| 4 | 3276 |
| 2 | 25479 |
| 0 | 36345 |

| DDT Info for S-Box 3 | |
|---|---|
| Values | Number of values in the difference table |
| 8 | 7 |
| 6 | 220 |
| 4 | 3333 |
| 2 | 25286 |
| 0 | 36179 |

| DDT Info for S-Box 4 | |
|---|---|
| Values | Number of values in the difference table |
| 8 | 9 |
| 6 | 222 |
| 4 | 3321 |
| 2 | 25296 |
| 0 | 36177 |

If initial state has non-zero input difference in one cell of the matrix, then after two round of Kalyna, all cells of the matrix have non-zero difference, this observation is same as AES.

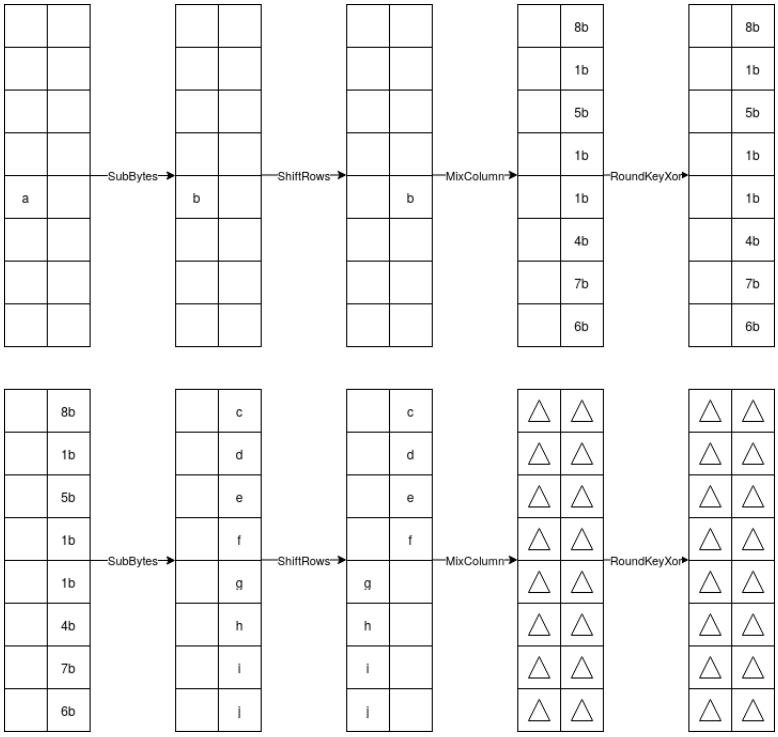


**Figure 3:** Difference propagation for two rounds of Kalyna

These are the difference of the Kalyna from the AES which makes Differential Cryptanalysis of AES inapplicable for Kalyna:

1. Use of four S-Boxes which depends on the row number of state matrix.
2. Use of Addition modulo $2^{64}$ operation, which affects the state matrix before $1^{st}$ round and in the last round of Kalyna.
3. Use of a new key expansion scheme that does not allow restoring the value of the source secret key from the value of one of the round keys.

## 6.2 Integral Cryptanalysis

In this analysis we have analysed 128-bit state with 128-bit key variant. We have checked how the *All* property and the *Constant* property propagates through the different states. Here $\Delta$ refers, the cell neither follow *Constant* property nor *All* property.

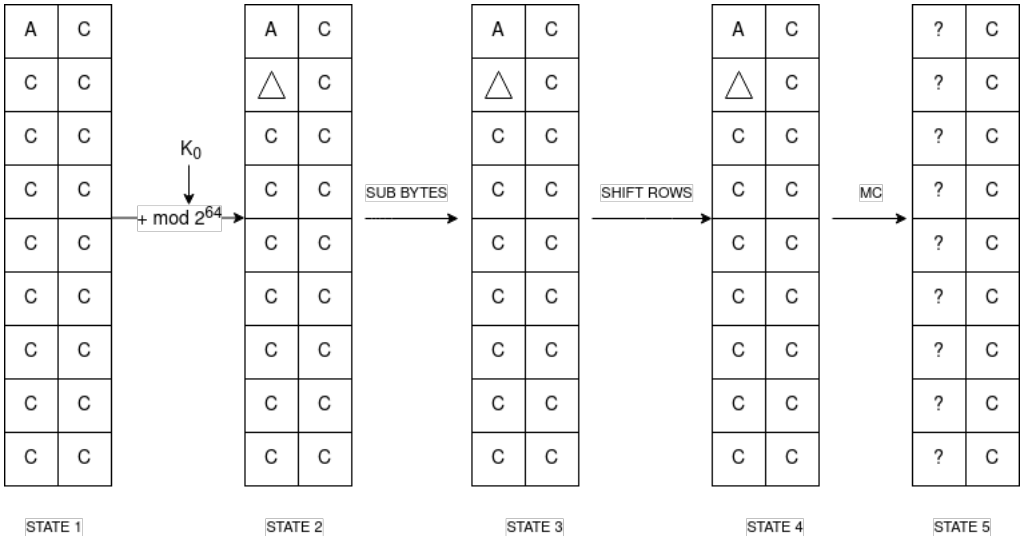


**Figure 4:** $All, Constant$ Property Propagation

#### 6.2.1 From State1 to State2: (add K mod $2^{64}$)

For our analysis we will choose the constant of $cell_{i\times 0}$ as 0x00, where $i \in 0, 1, ...,$(number of columns -1).

**Claim :** Passing from STATE1 to STATE2, at most property of the $cell_{1\times 0}$ changes (i.e. changes from $Constant$ to $\Delta$)

**Proof :** In the worst case, carry can be generated by choosing column as "FF00FFFFFFFFFFFFFFFFFFFFFFFFFF" and the $K_0$ = "FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF" in little endian format. Since the left most byte is represented as FF in both key and state column1, addition of both will create carry (i.e. 0xFF + 0xFF = 0x01FE). This carry cannot be propagated as we have choosen the $2^{nd}$ left byte as 0x00. This is also verified by the code written in python3.

#### 6.2.2 From State2 to State3: (Sub Bytes operation)

No change in any cell property, Since the sub-bytes of $All$ property gives $All$ property; sub-bytes of $Constant$ gives $Constant$ property; $\Delta$ gives $\Delta$ property.

#### 6.2.3 From State3 to State4: (Shift Row operation)

After this operation no property changes.

#### 6.2.4 From State4 to State5: (Mix Column operation)

**Claim :** After mix-column operation the $All$ property and the $Constant$ property of the cell got destroyed (i.e. after the mix column we can not say anything about the property of the cell anymore).

**Proof :**

$$\begin{bmatrix} 1 & 1 & 5 & 1 & 8 & 6 & 7 & 4 \\ 4 & 1 & 1 & 5 & 1 & 8 & 6 & 7 \\ 7 & 4 & 1 & 1 & 5 & 1 & 8 & 6 \\ 6 & 7 & 4 & 1 & 1 & 5 & 1 & 8 \\ 8 & 6 & 7 & 4 & 1 & 1 & 5 & 1 \\ 1 & 8 & 6 & 7 & 4 & 1 & 1 & 5 \\ 5 & 1 & 8 & 6 & 7 & 4 & 1 & 1 \\ 1 & 5 & 1 & 8 & 6 & 7 & 4 & 1 \end{bmatrix} \times \begin{bmatrix} i \\ \Delta \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{bmatrix} = \begin{bmatrix} 1i \oplus 1\Delta \oplus 5c_2 \oplus 1c_3 \oplus 8c_4 \oplus 6c_5 \oplus 7c_6 \oplus 4c_7 \\ 4i \oplus 1\Delta \oplus 1c_2 \oplus 5c_3 \oplus 1c_4 \oplus 8c_5 \oplus 6c_6 \oplus 7c_7 \\ 7i \oplus 4\Delta \oplus 1c_2 \oplus 1c_3 \oplus 5c_4 \oplus 1c_5 \oplus 8c_6 \oplus 6c_7 \\ 6i \oplus 7\Delta \oplus 4c_2 \oplus 1c_3 \oplus 1c_4 \oplus 5c_5 \oplus 1c_6 \oplus 8c_7 \\ 8i \oplus 6\Delta \oplus 7c_2 \oplus 4c_3 \oplus 1c_4 \oplus 1c_5 \oplus 5c_6 \oplus 1c_7 \\ 1i \oplus 8\Delta \oplus 6c_2 \oplus 7c_3 \oplus 4c_4 \oplus 1c_5 \oplus 1c_6 \oplus 5c_7 \\ 5i \oplus 1\Delta \oplus 8c_2 \oplus 6c_3 \oplus 7c_4 \oplus 4c_5 \oplus 1c_6 \oplus 1c_7 \\ 1i \oplus 5\Delta \oplus 1c_2 \oplus 8c_3 \oplus 6c_4 \oplus 7c_5 \oplus 4c_6 \oplus 1c_7 \end{bmatrix}$$

$$= \begin{bmatrix} 1i \\ 4i \\ 7i \\ 6i \\ 8i \\ 1i \\ 5i \\ 1i \end{bmatrix} \oplus \begin{bmatrix} 1\Delta \oplus 5c_2 \oplus 1c_3 \oplus 8c_4 \oplus 6c_5 \oplus 7c_6 \oplus 4c_7 \\ 1\Delta \oplus 1c_2 \oplus 5c_3 \oplus 1c_4 \oplus 8c_5 \oplus 6c_6 \oplus 7c_7 \\ 4\Delta \oplus 1c_2 \oplus 1c_3 \oplus 5c_4 \oplus 1c_5 \oplus 8c_6 \oplus 6c_7 \\ 7\Delta \oplus 4c_2 \oplus 1c_3 \oplus 1c_4 \oplus 5c_5 \oplus 1c_6 \oplus 8c_7 \\ 6\Delta \oplus 7c_2 \oplus 4c_3 \oplus 1c_4 \oplus 1c_5 \oplus 5c_6 \oplus 1c_7 \\ 8\Delta \oplus 6c_2 \oplus 7c_3 \oplus 4c_4 \oplus 1c_5 \oplus 1c_6 \oplus 5c_7 \\ 1\Delta \oplus 8c_2 \oplus 6c_3 \oplus 7c_4 \oplus 4c_5 \oplus 1c_6 \oplus 1c_7 \\ 5\Delta \oplus 1c_2 \oplus 8c_3 \oplus 6c_4 \oplus 7c_5 \oplus 4c_6 \oplus 1c_7 \end{bmatrix}$$

Since $\Delta$ is not constant, the *All* property is not preserved for column.

# 7 Conclusion

We have analyzed Kalyna with Differential and Integral Cryptanalysis. For Differential Cryptanalysis, we have observed that the differenence propagation for two round is same as AES. But the Differential Analysis of the AES is not applicable for Kalyna. This is because of the 4-sboxes, and $2^{64}$ modulo in the Kalyna. Though Integral Crypalysis for the One round is not successful, we have provided the reason for the faliure of the Integral Cryptanalysis. Which can be further used for improved integral attack. Throughout our analysis of Kalyna, we could not attack the cipher.

# 8 References

[1] [2][3][4] are our references.

# References

[1] In: (). DOI: https://en.wikipedia.org/wiki/Kalyna_(cipher).

[2] In: (). DOI: https://content.sciendo.com/downloadpdf.

[3] In: (). DOI: https://eprint.iacr.org/2015/650.pdf.

[4] In: (). DOI: https://eprint.iacr.org/2015/762.pdf.