Government Information Security Committee

ViEncrypt: a new block cipher for the Post-Quantum Cryptography Transition

Nguyen Bui Cuong, Nguyen Van Long, Nguyen Tuan Anh
Tran Sy Nam, Tran Duy Lai, Hoang Dinh Linh
Institute of Cryptographic Science and Technology
Government Information Security Committee, Hanoi, Vietnam
nguyenbuicuong@bcy.gov.vn

Outline

I. Introduction

II. Specification

III. Design strategy

IV. Security analysis

- ❑Classic cryptanalysis
- ❑Randomness Evaluation
- ❑Quantum resource estimation

V. Some implementation results

I. Introduction

ViEncrypt block cipher:

No.	Version	l	k	R
1.	ViEncrypt-128	128	128	6
2.			192	7
3.			256	8
4.	ViEncrypt-256	256	256	6
5.			384	7
6.			512	8

II. Specification

States

k_0^0	k_0^1	k_0^2	k_0^3	k_0^4	k_0^5	k_0^6	k_0^7
k_1^0	k_1^1	k_1^2	k_1^3	k_1^4	k_1^5	k_1^6	k_1^7
k_2^0	k_2^1	k_2^2	k_2^3	k_2^4	k_2^5	k_2^6	k_2^7
k_3^0	k_3^1	k_3^2	k_3^3	k_3^4	k_3^5	k_3^6	k_3^7

c)

k_0^0	k_0^1	k_0^2	k_0^3	k_0^4	k_0^5	k_0^6	k_0^7
k_1^0	k_1^1	k_1^2	k_1^3	k_1^4	k_1^5	k_1^6	k_1^7
k_2^0	k_2^1	k_2^2	k_2^3	k_2^4	k_2^5	k_2^6	k_2^7
k_3^0	k_3^1	k_3^2	k_3^3	k_3^4	k_3^5	k_3^6	k_3^7
k_4^0	k_4^1	k_4^2	k_4^3	k_4^4	k_4^5	k_4^6	k_4^7
k_5^0	k_5^1	k_5^2	k_4^3	k_5^4	k_5^5	k_5^6	k_4^7
k_6^0	k_6^1	k_6^2	k_6^3	k_6^4	k_6^5	k_6^6	k_6^7
k_7^0	k_7^1	k_7^2	k_7^3	k_7^4	k_7^5	k_7^6	k_7^7

d)

Key schedule state:
c) ViEncrypt-128
d) ViEncrypt-256

x_0^0	x_0^1	x_0^2	x_0^3
x_1^0	x_1^1	x_1^2	x_1^3
x_2^0	x_2^1	x_2^2	x_2^3
x_3^0	x_3^1	x_3^2	x_3^3

a)

x_0^0	x_0^1	x_0^2	x_0^3
x_1^0	x_1^1	x_1^2	x_1^3
x_2^0	x_2^1	x_2^2	x_2^3
x_3^0	x_3^1	x_3^2	x_3^3
x_4^0	x_4^1	x_4^2	x_4^3
x_5^0	x_5^1	x_5^2	x_4^3
x_6^0	x_6^1	x_6^2	x_6^3
x_7^0	x_7^1	x_7^2	x_7^3

b)

Internal state of the block cipher
a) ViEncrypt-128
b) ViEncrypt-256

II. Specification

Basic Transformation

SubCells and InvSubCells - Nonlinear Transformation

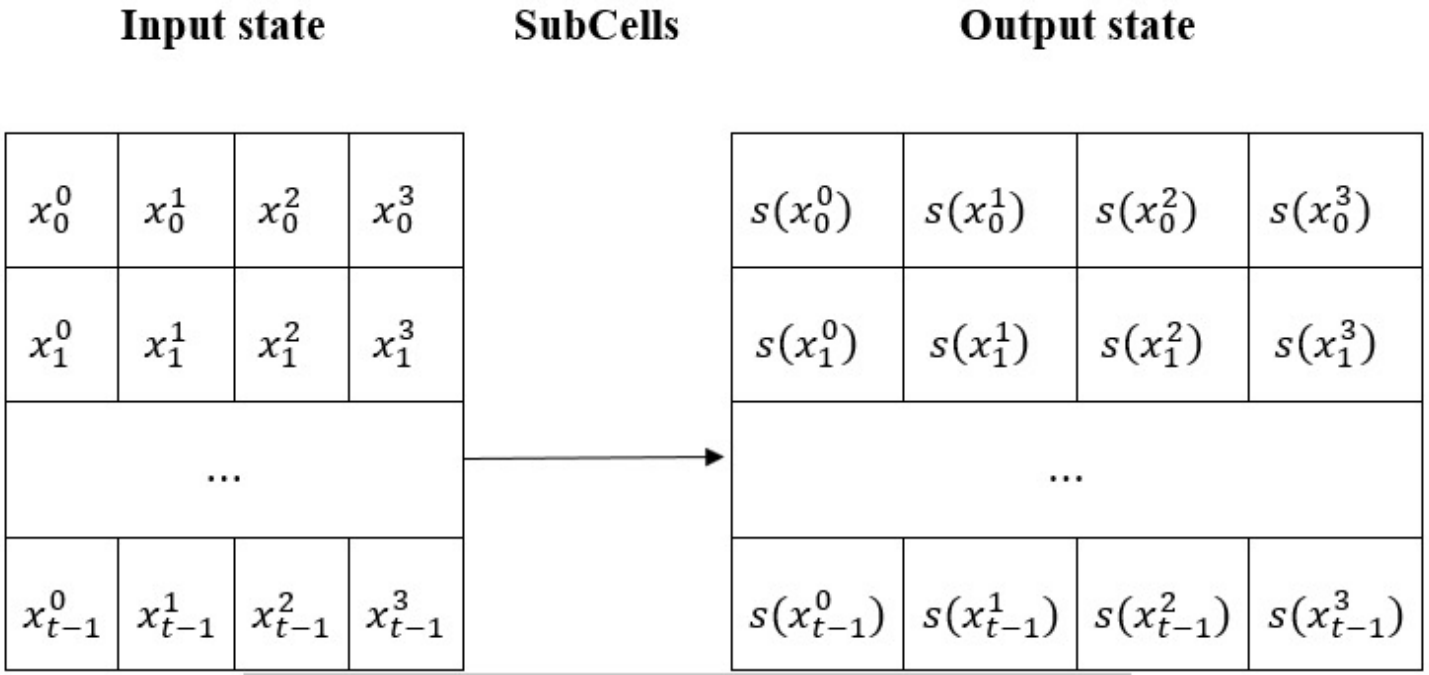


Figure 3: Illustration of SubCells transformation

II. Specification

Basic Transformation

MixWords and invMixWords: Linear Transformation

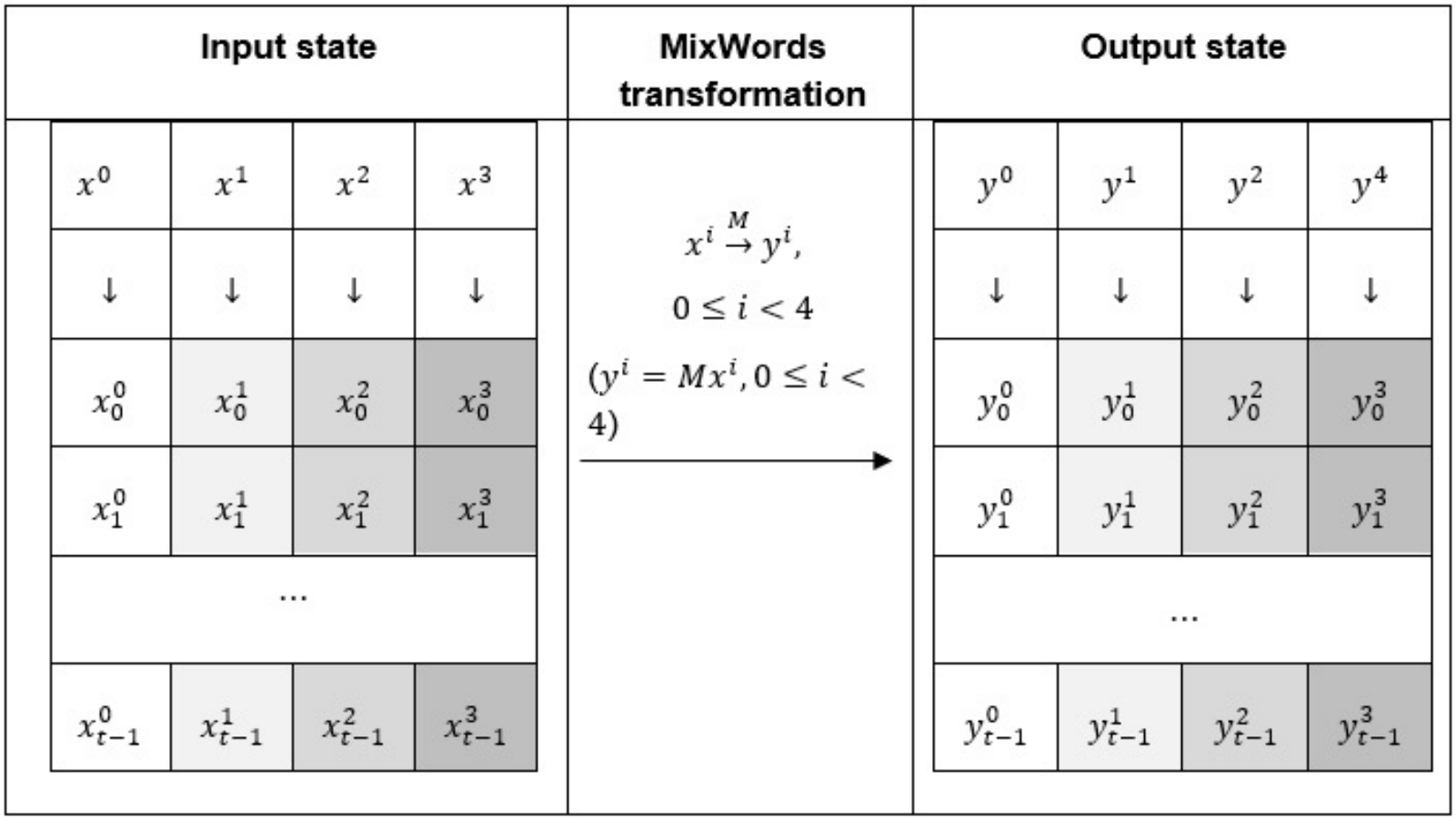


Figure 4 - The description of the MixWords transformation

II. Specification

Basic Transformation

Xwords – Linear Transformation

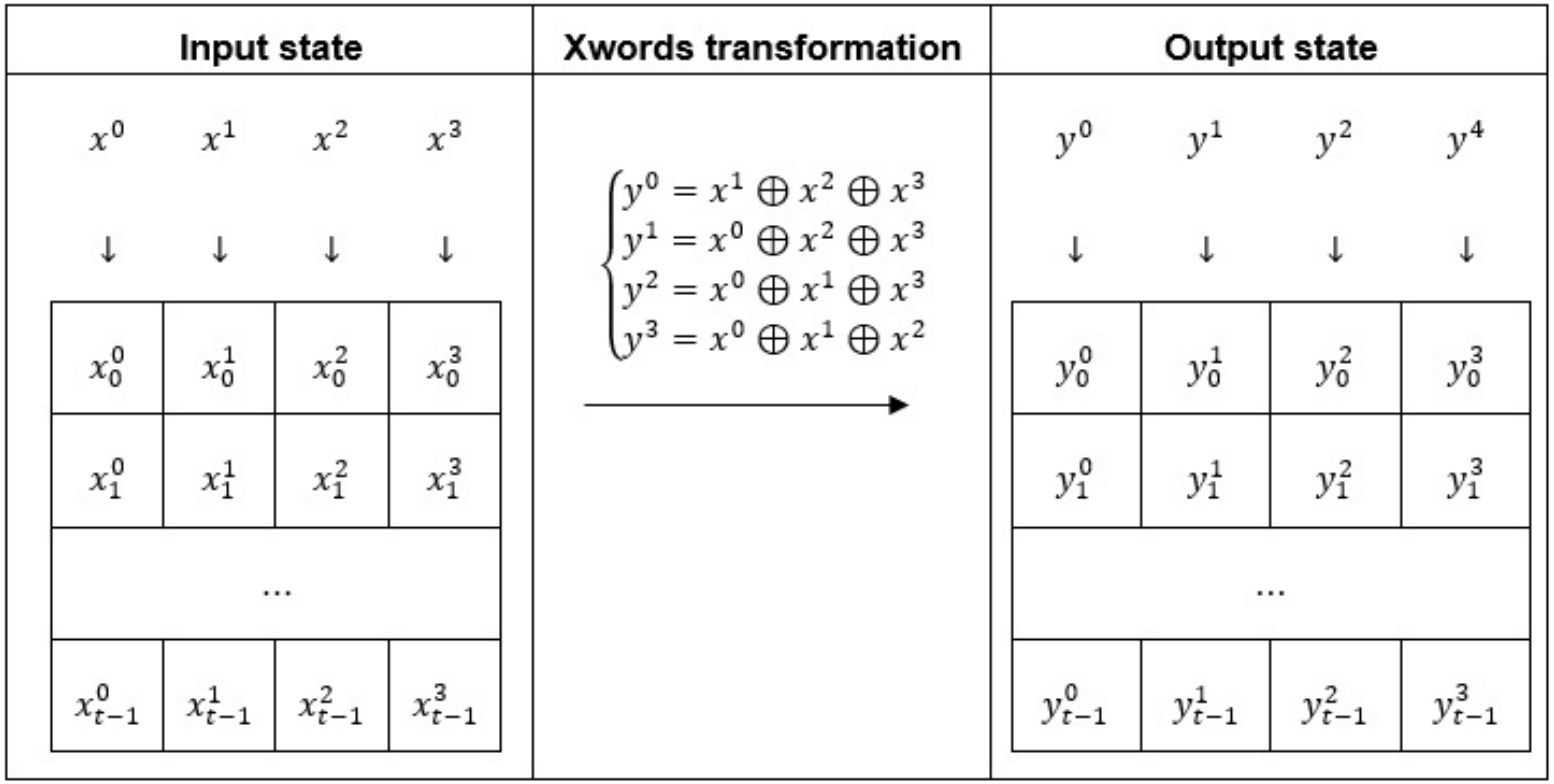
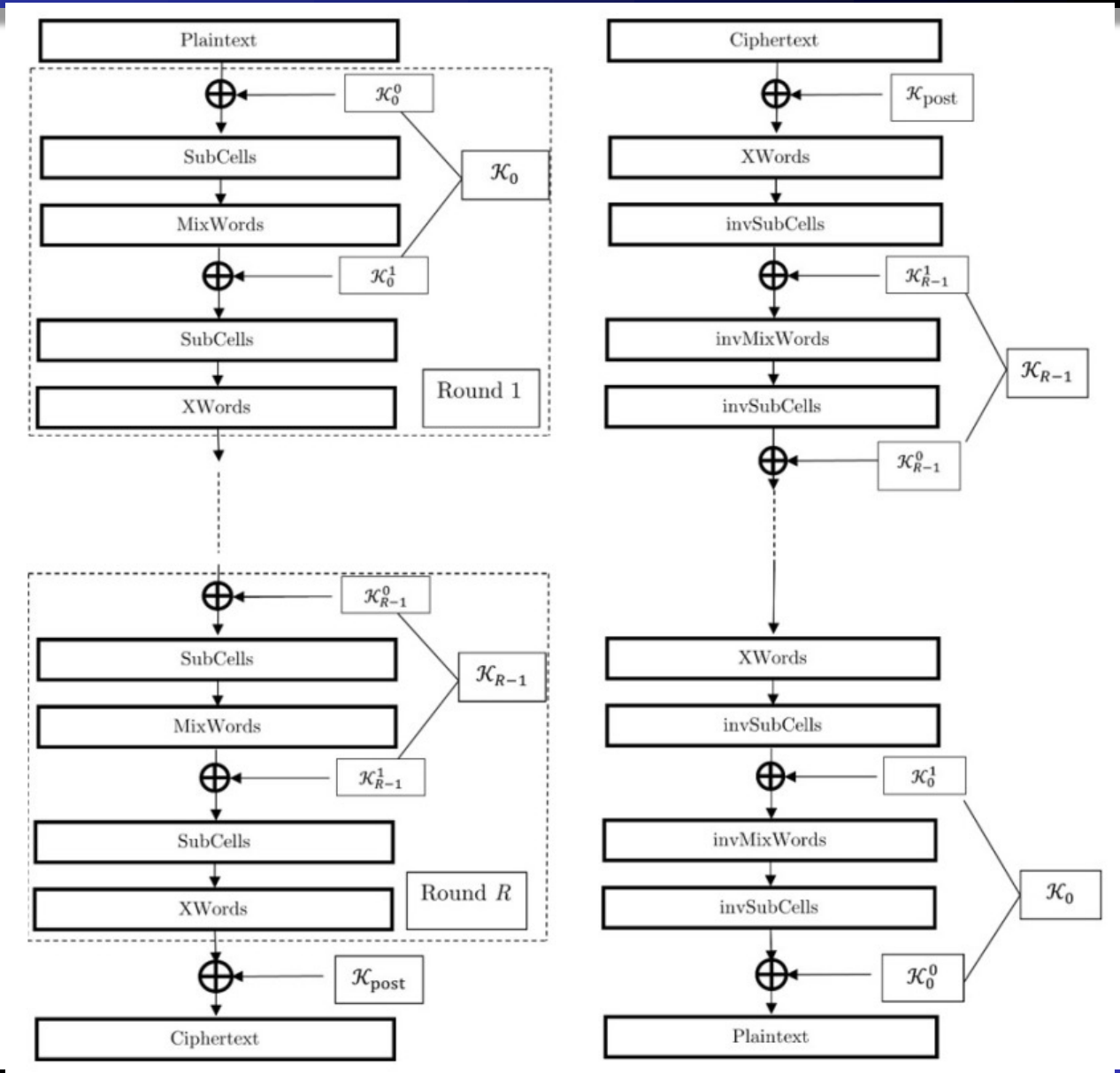


Figure 5 - The description of the Xwords transformation

II. Specification

Encryption and Decryption



II. Specification

Key Schedule

The key schedule of the ViEncrypt-l block cipher uses a key state transformation, denoted by UpdateKS$_l$. This transformation updates the key state of size of $2l$-bit $K = K^{\text{Left}} \| K^{\text{Right}} \in V_{2l}$ to get the next round key by the following steps:

- Initialization: The initial key state K_0 will be initialized from the master key $\mathcal{K}_{\text{master}}$ for the $2l$-bit key case and the additional key for the remaining case as follows:

$$K_0 = \begin{cases} \mathcal{k}^0 \| \dots \| \mathcal{k}^7 & \text{if } \mathcal{K}_{\text{master}} = \mathcal{k}^0 \| \dots \| \mathcal{k}^7 \in V_{2l} \\ \mathcal{k}^0 \| \dots \| k^4 \| \mathcal{k}^5 \| \overline{\mathcal{k}^2} \| \overline{\mathcal{k}^3} & \text{if } \mathcal{K}_{\text{master}} = \mathcal{k}^0 \| \dots \| k^5 \in V_{3l/2} \\ \mathcal{k}^0 \| \dots \| \mathcal{k}^3 \| \overline{\mathcal{k}^0} \| \dots \| \overline{\mathcal{k}^3} & \text{if } \mathcal{K}_{\text{master}} = \mathcal{k}^0 \| \dots \| \mathcal{k}^3 \in V_l \end{cases}$$

II. Specification

Key Schedule

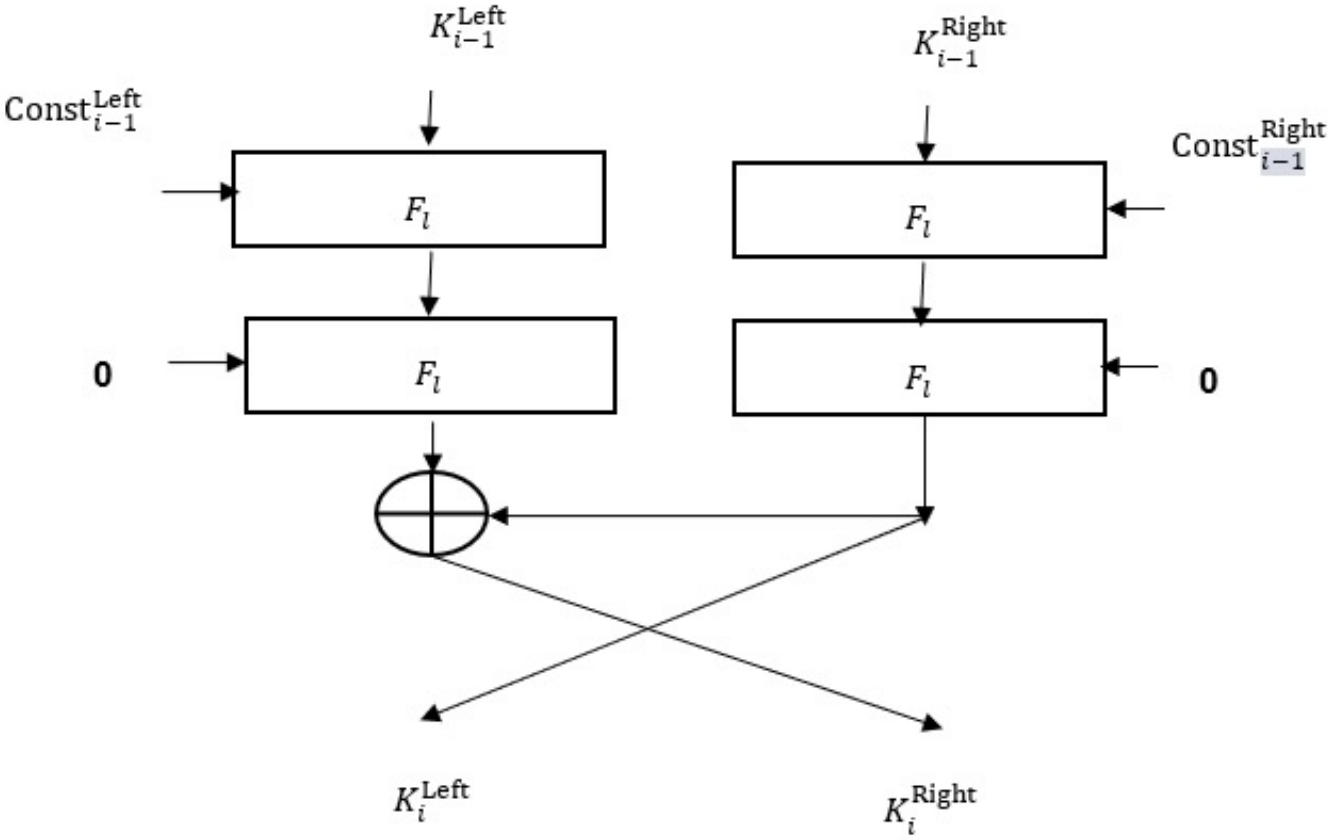


Figure 9 - $\text{UpdateKS}_l[C]$ transformation on the key state

III. Design strategy

FLC (Four-Leaf Clover) scheme

[1] Cuong Nguyen, Anh Nguyen, Phong Trieu, Long Nguyen, and Lai Tran. *Analysis of a new practically secure SPN-based scheme in the Luby-Rackoff model*. in *The 9th International Conference on Future Data and Security Engineering*. 2022. Springer.

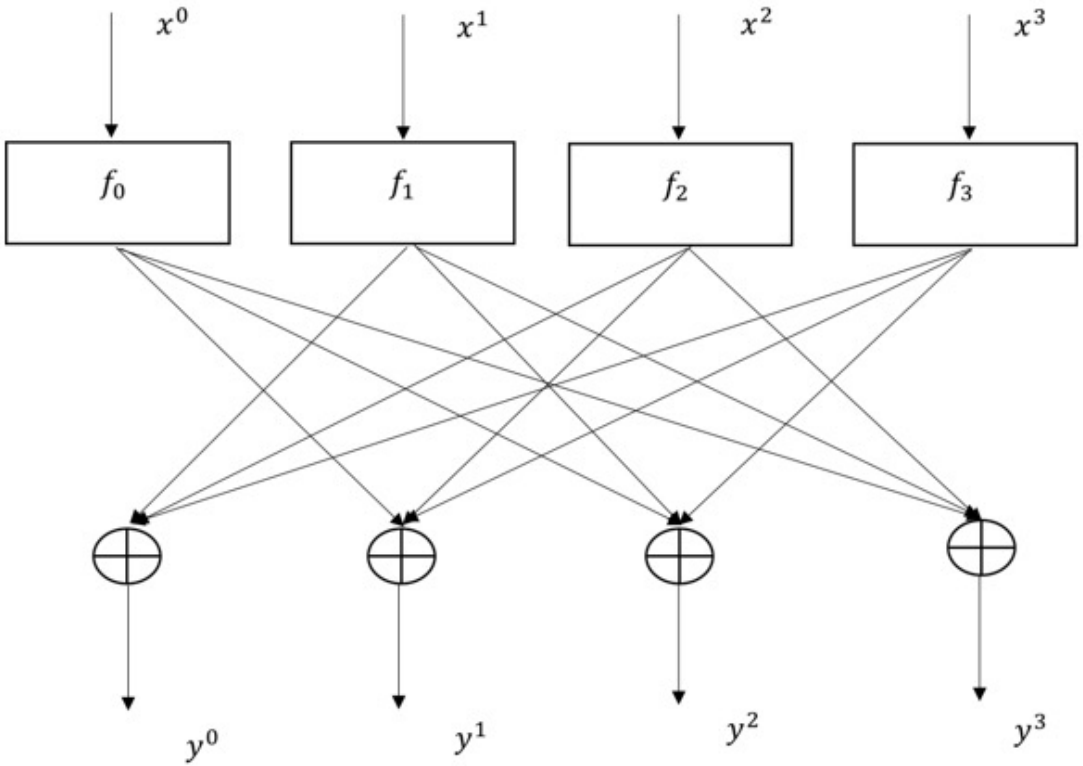


Fig. 10. 1-round FLC scheme.

III. Design strategy

Theorem 1. (Theorem 3, [1]) *Let 12 perfect random permutations* $f_0^1, f_1^1, f_2^1, f_3^1, f_0^2, f_1^2, f_2^2, f_3^2$, $f_0^3, f_1^3, f_2^3, f_3^3 \in P_w$. $C = \mathcal{F}^{(3)}(f_0^1, f_1^1, f_2^1, f_3^1, f_0^2, f_1^2, f_2^2, f_3^2, f_0^3, f_1^3, f_2^3, f_3^3) \in P_{4w}$ *be the 3-round FLC scheme, and* $F^* \in P_{4w}$ *is a perfect random permutation. For any pseudorandom distinguisher* $\mathcal{A}$ *allowed to make at most* d *encryption queries, we have*

$$\mathrm{Adv}_{\mathcal{A}}(C, F^*) \leq 5d(d-1)2^{-w+1}.$$

Theorem 2. (**Theorem 4, [1]**) *Let 20 perfect random permutations* $f_0^1, f_1^1, f_2^1, f_3^1, \dots, f_0^5, f_1^5, f_2^5, f_3^5 \in P_w$. $D = \mathcal{F}^{(5)}(f_0^1, f_1^1, f_2^1, f_3^1, \dots, f_0^5, f_1^5, f_2^5, f_3^5) \in P_{4w}$ *be the 5-round FLC scheme and* $F^* \in P_{4w}$ *is a perfect random permutation. For any super pseudorandom distinguisher* $\mathcal{A}$ *allowed to make at most* d *encryption and decryption queries, we have*

$$\mathrm{Adv}_{\mathcal{A}}(D, F^*) \leq d(d-1)2^{-w+4} + d(d-1)2^{-4w-1}.$$

III. Design strategy

S-box 8 bit s

[5] Denis Bonislavovich Fomin, "New classes of 8-bit permutations based on a butterfly structure", Математические вопросы криптографии, **10** (2019), 169–180.

Concretely, 8-bit S-box $s: \mathbb{F}_{2^8} \to \mathbb{F}_{2^8}$ takes an input $x = x_l \| x_r \in V_8$ and returns an output $y = y_l \| y_r \in \mathbb{F}_{2^8}$ where $x_l, x_r, y_l, y_r \in \mathbb{F}_{2^4}$ defined by the following formula:

$$y_r = \begin{cases} \pi_1(x_l) \cdot x_r \oplus 1, & x_r \neq 0, \\ \hat{\pi}_1(x_l) \oplus 1, & x_r = 0, \end{cases}$$

$$y_l = \begin{cases} \pi_2(x_r \cdot y_r), & y_r \neq 0, \\ \hat{\pi}_2(x_r), & y_r = 0, \end{cases}$$

where

$\pi_1 =$ {0x0, 0x1, 0xE, 0x9, 0xB, 0xD, 0x7, 0x6, 0x8, 0x3, 0xA, 0x4, 0xC, 0x5, 0x2, 0xF}

$\pi_2 =$ {0x0, 0x1, 0xD, 0xB, 0xE, 0x9, 0x6, 0x7, 0xA, 0x4, 0xF, 0x2, 0x8, 0x3, 0x5, 0xC}

$\hat{\pi}_1 =$ {0x0, 0x1, 0x9, 0xE, 0xD, 0xB, 0x7, 0x6, 0xF, 0x2, 0xC, 0x5, 0xA, 0x4, 0x3, 0x8}

$\hat{\pi}_2 =$ {0x0, 0x1, 0x9, 0xE, 0xD, 0xB, 0x7, 0x6, 0xF, 0x2, 0xC, 0x5, 0xA, 0x4, 0x3, 0x8}

III. Design strategy

Some cryptographic properties of 8-bit S-boxes in some ciphers

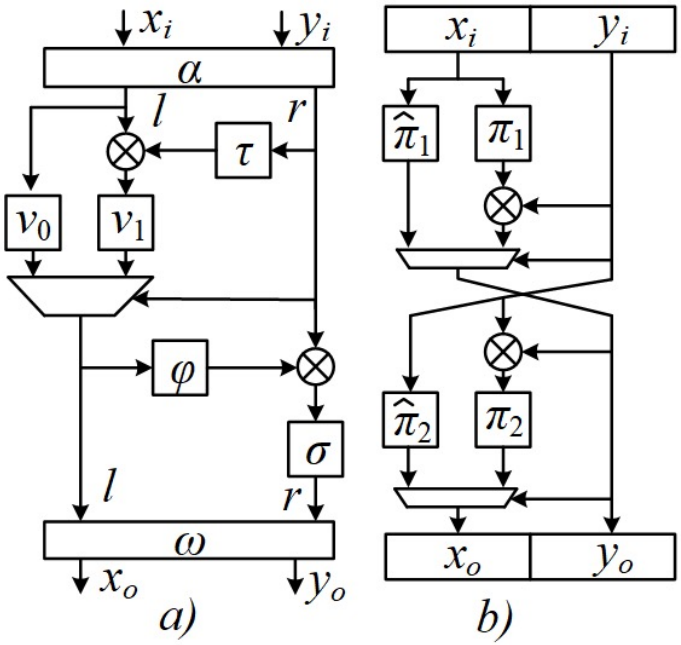
Cipher	ViEncrypt	Kalyna (pi1/pi2/ pi3/pi0)	Stree bog	AES	BelT	SMS4	Kuznye chik	ARIA _s2	Camellia	SEED _S1	E2	CLEFIA _S0
Permutation	**TRUE**											
Involution	**FALSE**											
Diff	6	8/8/8/8	8	4	8	4	8	4	4	4	10	10
DiffFreq	94	9/7/9/15	25	255	28	255	25	255	255	255	1	9
Diff1	4	6/4/6/15	4	2	4	2	4	4	4	2	4	0
CardD1	32	27	29	24	32	30	29	29	28	35	20	0
Lin	44	48	56	32	52	32	56	32	32	32	56	56
LinFreq	58	28	14	1275	10	1275	14	1275	1275	1275	9	52
Lin1	36	44	36	32	32	32	36	24	28	28	28	0
CardL1	59	58	59	60	60	61	59	60	60	57	51	0
max_degree	7	7	7	7	7	7	7	7	7	7	7	6
min_degree	7	7	7	7	6	7	7	7	7	7	6	6
MaxDegreeFreq	255	255	255	255	254	255	255	255	255	255	254	255
MinDegreeFreq	255	255	255	255	1	255	255	255	255	255	1	255
Max_ProductDegrees	**[7|7|7|7|7|7|7|8]**											[6|7|7|7|7|7|7|8]
LS_number	**0**											
max_y(y												

III. Design strategy

S-box 8 bit *s*

Nam, Tran Sy, Nguyen Van Long, and Nguyen Bui Cuong. "An Optimized Bit-Slice Implementation of Secure 8-Bit Sbox Based on Butterfly Structure." *2023 15th International Conference on Knowledge and Systems Engineering (KSE)*. IEEE, 2023.

TU decomposition of
Kuznyechik S-box – a) and our S-box – b)



III. Design strategy

The implementation complexity of S-boxes

S-box	Basic operations	Logic gates				Total	
		AND	OR	NOT	XOR		
Kuznyechik	Two multiplications $\otimes$ in $\mathbb{F}_{2^4}$	32			30	62	206
	α				9	9	
	ω				5	5	
	Multiplexer	4	3	1	8	16	
	τ	9	1	3	12	25	
	v_0	8	5	1	9	23	
	v_1	4	4	3	7	17	
	ϕ	9	3	3	10	25	
	σ	8	3	3	10	24	
ViEncrypt	Two multiplications $\otimes$ in $\mathbb{F}_{2^4}$	32			30	62	191
	Two multiplexers	8	6	2	16	32	
	π_1	7	3	1	14	25	
	π_2	8	4	1	12	25	
	$\hat{\pi}_1$	7	4		12	23	
	$\hat{\pi}_2$	7	4		12	23	
	Affine output				1	1	

III. Design strategy

MDS matrices

For ViEcnrypt-128, we consider the companion matrices of the form in [15] as follows:

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & L & 1 & L \oplus 1 \end{pmatrix}$$

Where its fourth power is the MDS matrix. With the base field $\mathbb{F}_{2^8}$ above, choose the linear transformation L as the multiplication with the element x (equal to 2 in the decimal system). Then, the companion matrices A and A^{-1} are determined as follows:

$$A_4 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 2 & 1 & 3 \end{pmatrix}, A_4^{-1} = \begin{pmatrix} 2 & 1 & 3 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Their fourth powers define the matrices used in MixWords and InvMixWords as follows $M_4 = A_4^4, M_4^{-1} = A_4^{-4}$.

III. Design strategy

ForViEncrypt-256, we choose the coefficients for the 8×8 companion matrix over the field $\mathbb{F}_{2^8}$, which has the following form:

$$A_8 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0\text{x}01 & 0\text{x}04 & 0\text{x}DB & 0\text{x}0C & 0\text{x}14 & 0\text{x}0C & 0\text{x}DB & 0\text{x}04 \end{pmatrix},$$

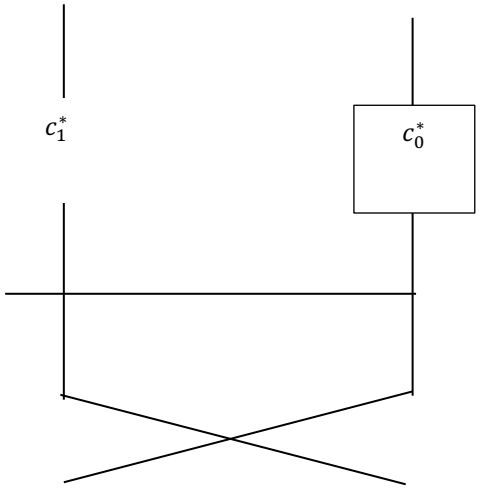
$$A_8^{-1} = \begin{pmatrix} 0\text{x}04 & 0\text{x}DB & 0\text{x}0C & 0\text{x}14 & 0\text{x}0C & 0\text{x}DB & 0\text{x}04 & 0\text{x}01 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

The eighth powers of these matrices give us an 8×8 MDS matrix over the field $\mathbb{F}_{2^8}$ with a primitive generating polynomial $f(x) = x^8 \oplus x^5 \oplus x^3 \oplus x \oplus 1$, i.e, $M_8 = A_8^8$ and $M_8^{-1} = A_8^{-8}$

III. Design strategy

Key schedule

1-round $\mathcal{V}$-scheme



[NBC2017] Bui Cuong Nguyen and Tuan Anh Nguyen, *Evaluating pseudorandomness and superpseudorandomness of the iterative scheme to build SPN block cipher.* Journal of Science and Technology on Information security, 2017. **40**(2): p. 40.

- The r-rounds V-scheme is indistinguishable from a perfect random function when r≥3.

- The r-rounds V-scheme is indistinguishable from a perfect random permutation when r≥5.

IV. Security analysis

❑Classic cryptanalysis

Differential and linear cryptanalysis

Corollary 1. *Any trail over two rounds of ViEcnrypt-128 has at least 20 differential/linear active S-boxes.*

Corollary 2. *Any trail over two rounds of ViEncrypt-256 has at least 36 differential/linear active S-boxes.*

IV. Security analysis

- Classic cryptanalysis

ViEncrypt-128

Round	Minimum active S-boxes number	Differential cryptanalysis		Linear cryptanalysis	
		The Upper bound of probability	Complexity	Upper bound of probability	Complexity
1	5	$2^{-27,085}$	$2^{27,085}$	$2^{-12,7}$	$2^{25,4}$
2	20	$2^{-108,34}$	$2^{108,34}$	$2^{-50,8}$	$2^{101,6}$
3	25	$2^{-135,425}$	$2^{135,425}$	$2^{-63,5}$	2^{127}
4	40	$2^{-216,68}$	$2^{216,68}$	$2^{-101,6}$	$2^{203,2}$
5	45	$2^{-243,765}$	$2^{243.765}$	$2^{-114,3}$	$2^{228,6}$
6	60	$2^{-325,02}$	$2^{325,02}$	$2^{-152,4}$	$2^{304,8}$
7	65	$2^{-352,105}$	$2^{352,105}$	$2^{-165,1}$	$2^{330,2}$
8	80	$2^{-433,36}$	$2^{433,36}$	$2^{-203,2}$	$2^{406,4}$

IV. Security analysis

- Classic cryptanalysis

ViEncrypt-256

Round	Minimum active S-boxes number	Differential cryptanalysis		Linear cryptanalysis	
		The Upper bound of probability	Complexity	Upper bound of probability	Complexity
1	9	$2^{-48,753}$	$2^{48,753}$	$2^{-22,86}$	$2^{45,72}$
2	36	$2^{-195.012}$	$2^{195.012}$	$2^{-91,44}$	$2^{182,88}$
3	45	$2^{-243,765}$	$2^{243,765}$	$2^{-114,3}$	$2^{228.6}$
4	72	$2^{-390,024}$	$2^{390,024}$	$2^{-182,88}$	$2^{365,76}$
5	81	$2^{-438.777}$	$2^{438.777}$	$2^{-205,74}$	$2^{411,48}$
6	108	$2^{-585.036}$	$2^{585.036}$	$2^{-274,32}$	$2^{548,64}$
7	117	$2^{-633,789}$	$2^{633,789}$	$2^{-297,18}$	$2^{594,36}$
8	144	$2^{-780,048}$	$2^{780,048}$	$2^{-365,76}$	$2^{731,52}$

IV. Security analysis

❑Classic cryptanalysis

Boomerang attack:

- **ViEncrypt-128: 3 rounds – complexity 2^{270}**
- **ViEncrypt-256: 3 rounds – complexity 2^{487}**

Integral attack:

- **ViEncrypt-128: 3,5 rounds – complexity 2^{265}**
- **ViEncrypt-256: 3,5 rounds – complexity 2^{521}**

Algebraic attack:

Cipher	No.of variables	No. of equations		
		linear	quadratic	of degree 3
ViEncrypt-128/128	13,312	9,600	0	176,400
ViEncrypt-128/192	15,872	11,520	0	211,680
ViEncrypt-128/256	18,432	13,440	0	246,960
ViEncrypt-256/256	26,624	19,200	0	352,800
ViEncrypt-256/384	31,744	23,040	0	423,360
ViEncrypt-256/512	36,864	26,880	0	493,920

Differential related key attack:

- **ViEncrypt-128: 4 rounds – complexity 2^{541}**
- **ViEncrypt-256: 3 rounds – complexity 2^{975}**

IV. Security analysis

❑Classic cryptanalysis

Other cryptanalysis

We have also considered the security of other cryptanalyses such as, linear-differential cryptanalysis, impossible differential cryptanalysis, zero correlation cryptanalysis, invariant subspace cryptanalysis, etc. In particular, for invariant subspace cryptanalysis, due to the FLC structure, the invariant spaces on the V_l of ViEncrypt-l will be reduced to the invariant spaces on the V_w space in each substate. However, in each substate, linear layers based on MDS matrices of full-block size are used, resulting in no real invariant subspace to exploit in cryptanalysis. For impossible differential cryptanalysis, we analyzed an impossible difference based on possible mid-errors due to linear transformations MixWords and XWords. Furthermore, these differences are also considered to extend the attack to a more significant number of rounds. Nevertheless, it is no longer practical for ViEncrypt with number of round greater than three.

IV. Security analysis

❑Randomness Evaluation

We have performed a randomness evaluation for variants of ViEncrypt round-by-round according to a pre-selection process on four non-random input Plaintext sets (Low-Density Plaintext (LW) data set, High-Density Plaintext (HW) data set, 1-Bit Plaintext Avalanche data set (Av1), Plaintext Rotation (Rot) data set), using seven appropriate statistical tests (Frequency, Run, Longest of 1 in a block, Serial, Approximate Entropy, Cumulative Sum). Then, we have the evaluation results through the rounds summarized:

Algorithm	# Rounds	# Rounds Random	Data Sets
ViEncrypt-256/512	8	≥ 2	LW, HW, Av1, Rot
ViEncrypt-256/384	7	≥ 2	LW, HW, Av1, Rot
ViEncrypt-256/256	6	≥ 2	LW, HW, Av1, Rot
ViEncrypt-128/256	8	≥ 2	LW, HW, Av1, Rot
ViEncrypt-128/192	7	≥ 3	LW, HW, Av1, Rot
ViEncrypt-128/128	6	≥ 2	LW, HW, Av1, Rot

IV. Security analysis

❑Quantum resource estimation

Table 8: Quantum Resources for ViEncrypt without Ancilla Qubits

Cipher	No. of Qbit	CNOT gates	8-Toffoli gates	X gates	Swap gates
ViEncrypt-128/128	384	916,960	229,440	1,019,779	768
ViEncrypt-128/192	384	1,069,680	267,680	1,189,660	896
ViEncrypt-128/256	384	1,222,400	305,920	1,359,543	1,024
ViEncrypt-256/256	768	1,855,760	458,880	2,039,427	1,536
ViEncrypt-256/384	768	2,164,840	535,360	2,379,164	1,792
ViEncrypt-256/512	768	2,473,920	611,840	2,718,903	2,048

Table 9: Quantum Resources for ViEncrypt using Ancilla Qubits

Cipher	No. of Qbits	CNOT gates	4-Toffoli gates	Toffoli gates	X gates	Swap gates	Full depth
ViEncrypt-128/128	10,944	77,920	4,800	69,120	16,579	768	3,265
ViEncrypt-128/192	12,704	90,800	5,600	80,640	19,260	896	3,805
ViEncrypt-128/256	14,464	103,680	6,400	92,160	21,943	1,024	4,346
ViEncrypt-256/256	21,888	177,680	9,600	138,240	33,027	1,536	3,612
ViEncrypt-256/384	25,408	207,080	161,280	161,280	38,364	1,792	4,210
ViEncrypt-256/512	28,928	236,480	12,800	184,320	43,703	2,048	4,809

IV. Security analysis

❑Quantum resource estimation

Table 10: Complexity of ViEncrypt against exhaustive key search attacks using the Grover algorithm (r- number of plaintext/ciphertext pairs)

Cipher	r	No. of qubit (M)	Total gates (G)	Full depth (FD)	FD-G Complexity (FD×G)	FD-M Complexity (FD×M)
ViEncrypt-128/128	1	10,944	1.40×2^{80}	1.25×2^{76}	1.75×2^{156}	1.66×2^{89}
ViEncrypt-128/192	2	12,704	1.65×2^{112}	1.46×2^{108}	1.21×2^{223}	1.75×2^{123}
ViEncrypt-128/256	2	14,464	1.86×2^{145}	1.67×2^{141}	1.55×2^{287}	1.82×2^{155}
ViEncrypt-256/256	1	21,888	1.44×2^{145}	1.39×2^{140}	1.00×2^{287}	1.02×2^{154}
ViEncrypt-256/384	2	25,408	1.67×2^{210}	1.61×2^{205}	1.39×2^{416}	1.33×2^{220}
ViEncrypt-256/512	2	28,928	1.32×2^{276}	1.84×2^{269}	1.21×2^{546}	1.34×2^{284}

AES128	1	3.257	$1{,}17 \cdot 2^{82}$	$1{,}27 \cdot 2^{74}$	$1{,}49 \cdot 2^{156}$	$1{,}01 \cdot 2^{86}$
AES192	2	7.161	$1{,}29 \cdot 2^{115}$	$1{,}64 \cdot 2^{106}$	$1{,}03 \cdot 2^{222}$	$1{,}43 \cdot 2^{119}$
AES256	2	7.537	$1{,}84 \cdot 2^{147}$	$1{,}16 \cdot 2^{139}$	$1{,}07 \cdot 2^{287}$	$1{,}94 \cdot 2^{151}$

Some implementation results

❖ Software implementation

We implemented ViEncrypt in software by using the precomputed lookup tables implementation and bit-sliced implementation methods. The cipher was running in ECB mode of operation. The source code is written in C++ and optimized for 64-bit hardware platforms with implementation methods using precomputed lookup tables. The code does not contain any assembler instructions. It neither uses any special registers nor invocation of SIMD commands. The code was compiled in the Visual Studio 2022 environment for the x64/Intel platform. The measurements were performed on a single Intel i7- 2600 core @ 3,4GHz processor in Windows 7 OS. The speed of our implementation is measured in Megabits per second. The results are given in.

Some implementation results

- **Software implementation using lookup tables**

No.	Cipher	Size of lookup tables (enc + dec) in KBytes	Ecn. speed (Mb/s)	Dec. speed (Mb/s)	Ref.
128-bit block size					
1	ViEncrypt-128/128	8	**1529**	**1518**	Ours
	ViEncrypt-128/192	8	**1320**	**1299**	
	ViEncrypt-128/256	8	**1156**	**1151**	
2	AES-128/128	8	1783	1779	Gladman
	AES-128/192	8	1526	1520	
	AES-128/256	8	1323	1315	
3	Kuznyechik	128	843	804	Oliynykov
4	Kalyna-128/128	32	1749	1716	
	Kalyna-128/256	32	1331	1329	
256-bit block size					
1	ViEncrypt-256/256	16	**1599**	**1612**	Outs
	ViEncrypt-256/384	16	**1386**	**1390**	
	ViEncrypt-256/512	16	**1225**	**1226**	
2	Kalyna-256/256	16	1616	n/a	Oliynykov
	Kalyna-256/512	16	1299	n/a	

Some implementation results

❖ Hardware implementation

We implemented ViEncrypt on a Kintex-7 FPGA chip (xc7k160tfbg676-3) using Vivado 2015. The implementation technique is iterative looping, with each round requiring three clock cycles or 18 clock cycles for six rounds. The key schedule is pre-computed on the fly, providing round keys for both encryption and decryption processes. The Mbps/Slice parameter is used to evaluate implementation efficiency. The results are given as follow:

Cipher	Slice	Frequency (MHz)	Clock	Speed (Mbits/s)	Mbps/Slice
ViEncrypt-128/128	370	309	18	2,197	5.93
ViEncrypt-128/192	387	309	21	1,883	4.86
ViEncrypt-128/256	393	309	24	1,648	4.19
ViEncrypt-256/256	1791	250	18	3,555	1.98
ViEncrypt-256/384	1863	250	21	3,047	1.63
ViEncrypt-256/512	1896	250	24	2,666	1.4

Thank you for listening!