High-Performance Reliable Block Encryption Algorithms Secured against Linear and Differential Cryptanalytic Attacks

Sergiy Gnatyuk[1], Vasyl Kinzeryavyy[1], Maksim Iavich[2], Dmytro Prysiazhnyi[3], Khalicha Yubuzova[4]

[1]National Aviation University, Kyiv, Ukraine
[2]Scientific Cyber Security Association, Tbilisi, Georgia
[3]Vinnytsia National Technical University, Vinnytsia, Ukraine
[4]Satbayev University, Almaty, Kazakhstan
s.gnatyuk@nau.edu.ua v.kinzeryavyy@nau.edu.ua m.iavich@scsa.ge
dimpris@gmail.com hali4a@mail.ru

Abstract. To be secure, modern information and communication technology (ICT) needs reliable encryption. Symmetric cryptography combines encryption algorithms that use the same cryptographic keys for both encryption of plaintext and decryption of ciphertext. These algorithms are used for confidentiality ensuring in different spheres but most of them are worn out and outdated. Modern encryption algorithms development is very actual task for ICT (up to date and next generation). In this paper to improve the security effectiveness of electronic information resources, two encryption algorithms have been developed based on fixed lookup tables with extended-bit depth and dynamic key-dependent lookup tables. The developed algorithms are at least two times faster than the previous national encryption standard and virtually secure to linear and differential cryptanalysis. Properties of random sequences formed using the proposed algorithms' encryption (in counter mode) were explored in the environment of NIST STS statistical tests, according to which they passed integrated control by mentioned tests with better results than other generators.

Keywords: IT security, confidentiality, cryptography, cipher, block encryption algorithm, cryptographic security, reliable encryption, algebraic coding theory.

1. Introduction

As a result of modern global challenges in information security on the state level, the requirements for state information resources security and other critical information (transferred and stored in information and communication systems) are constantly growing [1-2]. The cryptographic security of these resources is one of the most significant security measures (especially in critical information infrastructure protection [2-3]). Cryptographic security is one of the most reliable and effective methods of information security; its principal and undeniable advantage is the data protection without access to it. The principal criterion for choosing cryptosystems is security, but for some applications (e.g., Big Data encryption, online banking payment systems etc.) the key role played by cryptographic data processing is speed.

Despite the wide variety of modern encryption methods and algorithms, not all have the necessary level of effectiveness (speed and security). In addition, the rapid development of computational tools and their simultaneous cost reduction have led to new requirements for both the security and the performance of cryptosystems–old cryptographic algorithms disappear, and new ones must pass specific competitive selection and prove their ability to provide security for a specific period of time in the future [4-10]. Algorithm 28147:2009 was Ukrainian National Encryption Standard until 2014. DSTU 7624:2014, the new Encryption Standard since 2015, has not been well investigated [6, 9]. However, considering the significant progress in the field of cryptanalysis methods and tools, Ukrainian National Encryption Standard 28147:2009 has become obsolete [9]. Thus, in accordance with the requirements of the New European Schemes for Signatures, Integrity, and Encryption (NESSIE) project [5], the algorithm of Ukrainian Encryption Standard 28147-2009 applies only to the third (lowest) class of security [8]. In addition, previous Ukrainian National Encryption Standard 28147-2009 does not satisfy the modern requirements for data encryption speed [7] that is important for mentioned tasks.

Additional disadvantages include complexity of hardware implementation and use of secret long-term key data, which are supplied in a prescribed manner [7]. Therefore, the development of new algorithms to improve the efficiency of information security is an important scientific task. The purpose of this work is to contribute to improvement in the efficiency of electronic information resources security through the development of modern secure high-performance reliable block ciphers.

2. Method for Speed of Block Ciphers Increasing

Consider a class of block ciphers with a set of open (encrypted) messages $V_n = \{0,1\}^n$, $n = 128 \cdot p$, $p \in N$, a set of round keys $K = V_n$, and a family of cryptographic transformations $F_k = f_{r,k_r} \circ ... \circ f_{1,k_1}$, $k = (k_1, ..., k_r) \in K^r$, where r is the number of encryption rounds.

The round transformation $f_{i,k}(x)$ for any $x \in V_n$, $k \in K$, and $i \in \overline{1,r}$ is described as $f_{i,k} = \begin{cases} \varphi(x+k), & i < r \\ s_r(x+k), & i = r \end{cases}$, where + is defined separately for each round operation of addition modulo 2 or $2^{32 \cdot l}$, $l \in N$, $l \le n/32$. Substitutions $\varphi(x)$ and $s_i(x)$ are defined by the formulas $\varphi(x) = L(s_i(x))$, $x \in V_n$, $s_i(x) = (s_i'(x_{c-1}), ..., s_i'(x_0))$, where $x = (x_{c-1}, ..., x_0)$, $x_j \in V_t$, $t \ge 4$, $c = n/t$, $j \in \overline{0, c-1}$, s_i' involves m lookup tables on the set V_t, used in the i-th round ($m \in \overline{1,c}$), and $L(x)$ is a linear transformation used in a block cipher. s_i' – one of m ($m \in N$) substitution tables on set V_t, that is using in i-th round, $L(x)$ – linear transformation using in block cipher.

For the above class of block ciphers there are fairly well known analytical upper estimates of the parameters [11, 12], characterizing practical security against linear [13, 15] and differential [13, 14] cryptanalysis attacks:

$$EDP(\Omega) \le \Delta^{(r-1)B_L/2+1} \quad (1)$$

$$ELP(\Omega) \le \Lambda^{(r-1)B_L/2+1} \quad (2)$$

where $\Delta = \max\left\{d_+^{s'}(\alpha,\beta) : \alpha,\beta \in V_t \setminus \{0\}\right\}$ is the maximum probability difference of the substitution table s' [11], $\Lambda = \max\left\{l_+^{s'}(\alpha,\beta) : \alpha,\beta \in V_t \setminus \{0\}\right\}$ is the maximum probability of linear approximation of the substitution table s' [11], B_L is the number of the revitalization branches of the following linear transformation $L(x)$ ($B_L = \min\left\{wt(x) + wt\left(xL^{-1}\right)\right\}$) [11], $EDP(\Omega)$ is the average differential characteristics probability of Ω [11], and $ELP(\Omega)$ is the average linear characteristic probability of Ω [11].

Variables $d_+^{s'}(\alpha,\beta)$ and $l_+^{s'}(\alpha,\beta)$ are defined by the following formulas (if the operation "+" is addition modulo 2) [11, 12]:

$$d_+^{s'}(\alpha,\beta) = 2^{-t}\sum_{k\in V_t}\delta(s'(k+\alpha)\oplus s'(k),\beta), \quad (3)$$

$$l_+^{s''}(\alpha,\beta) = 2^{-t}\sum_{k\in V_t}\left(2^{-t}\sum_{x\in V_t}(-1)^{\alpha x\oplus\beta s'(x+k)}\right)^2 \quad (4)$$

where δ is the Kronecker delta symbol, $\delta(u,v) = \begin{cases}0, u <> v\\ 1, u == v\end{cases}$.

According to the standard methodology of constructing block ciphers [16], using linear transformations with a large parameter B_L and substitution tables with smaller indices Δ and Λ reasonably decrease the number of encryption rounds of a block cipher r, ensuring its practical security against of linear and differential cryptanalysis. The minimum number of rounds (see Table 1) was determined on the basis of formulas (1) and (2), yielding practical resistance to linear and differential cryptanalysis of block ciphers (considered class), where the length of the secret key and data block is 128 bits.

Table 1. Minimum number of rounds for ensuring practical security against linear and differential cryptanalysis

MDS-codes, covering bytes	Substitution table on the set		
	V_8 ($\Delta = \Lambda = 2^{-6}$)	V_{16} ($\Delta = \Lambda = 2^{-14}$)	V_{32} ($\Delta = \Lambda = 2^{-30}$)
4 bytes ($B_L = 5$)	$r = 10$	$r = 5$	$r = 3$
8 bytes ($B_L = 9$)	$r = 6$	$r = 3$	$r = 2$
16 bytes ($B_L = 17$	$r = 4$	$r = 2$	$r = 2$

As a linear transformation of the considered maximum distance separable (MDS) codes, covering w bytes ($w = 4, 8, 16$) allows the number of activation branches $B_L = w+1$. We investigated the substitution table on the set V_q, where $q = 8, 16, 32$, for which the theoretically achievable levels of Δ and Λ are equal to $2^{-(q-2)}$.

According to Table 1, the performance of block ciphers (considered as a class) can be improved by taking the following steps:

1) Expand the variety of permutations to use the substitution table on the set V_{16} (most modern cryptographic algorithms use lookup tables on multiples of V_8). Using such a table requires only 128 KB of memory, which is now acceptable. A substitution table on the set V_{32} requires much more memory, resulting in it being impractical for current use.

2) Replace some MDS codes to cover a larger number of bytes (this will increase the number of operations per round, but not significantly).

3) Reduce the number of block cipher rounds to improve performance.

For example, this method can be applied to block cipher Kalyna encryption algorithms ($\Delta, \Lambda \geq 2^{-5}$, $B_L = 9$, $r = 11$) [11] and the Advanced Encryption Standard (AES) ($\Delta = \Lambda = 2^{-6}$, $B_L = 5$, $r = 11$) [4], but in this case (with decreasing r), investigating their security in regard to other methods of cryptanalysis is a very difficult task. However, the opportunity to enhance their performances deserves attention.

3. New Block Encryption Algorithms Development

On the basis of the described method for increasing the speed of block ciphers, two algorithms have been developed for encrypting the information using a fixed table lookup with an extended bit depth *(Luna)* and with dynamic key-dependent lookup tables *(Neptune)*. The pseudocode for the algorithms is shown in Fig. 1a and 1b, respectively.

Luna

Input: 128-bit input data block $state$,

128-bit extended keys $subkey[i]$, $i = \overline{0, r+2}$.

Output: 128-bit output data block.

1. $AddKeyMod2(state, subkey[1])$;
2. $For \quad j = 0, \quad j < r-1, \quad j{+}{+} \quad do$
2.1. $SubBytes_{Luna}(state)$;
2.2. $ShiftRows(state)$;
2.3. $\mathrm{MixColumns}(state)$;
2.4. $AddKeyMod2(state, subkey[j+2])$;
3. $SubBytes_{Luna}(state)$;
4. $ShiftRows(state)$;
5. $AddKeyMod2(state, subkey[r+1])$;
6. *return state* ;

a

Neptune

Input: 128-bit input data block $state$,

128-bit extended keys $subkey[i]$, $i = \overline{0, 2 \cdot r}$.

Output: 128-bit output data block.

1. $AddKeyMod2(state, subkey[0])$;
2. $For \quad j = 0, \quad j < r-1, \quad j{+}{+} \quad do$
2.1. $SubBytes_{\mathrm{Neptun}}(state, subkey[2 \cdot j + 1])$;
2.2. $ShiftRows(state)$;
2.3. $\mathrm{MixColumns}(state)$;
2.4. $AddKeyMod2(state, subkey[2 \cdot j + 2])$;
3. $SubBytes_{\mathrm{Neptun}}(state, subkey[2 \cdot r - 1])$;
4. $ShiftRows(state)$;
5. $AddKeyMod2(state, subkey[2 \cdot r])$;
6. *return state* ;

b

Fig. 1. Pseudocode for encrypting the (a) Luna and (b) Neptune encryption algorithms

These algorithms use 128-bit data blocks (represented as 4×4 byte matrices) with secret-key lengths of 128, 256, and 512 bits, formed from the required number of 128-bit extended keys represented as matrices of size 4×4 bytes. The number r of encryption rounds depends on the length of the secret key. With secret-key lengths of 128, 256, and 512 bits, $r=7$, 9, 13 in Luna and $r=9$, 13, 21 in Neptune, respectively.

The operation $AddKeyMod2\left(state, subkey\left[i\right]\right)$ is bitwise addition modulo 2 of the corresponding bits of the extended $subkey\left[i\right]$ key and a $state$ data block.

The $\mathrm{MixColumns}\left(state\right)$ operation is a linear $state$ sequence transformation. In this operation, the $state$ data block is split into two parts of eight bytes (the first two four-byte columns form one eight-byte part, and the other eight – the second part), each of which is considered as a polynomial over the field $GF\left(2^8\right)$ with eight terms, which are multiplied by x^8+1 modulo a fixed polynomial $c\left(x\right)$ (see Fig. 2), thereby ensuring that the number of activation branches is nine. The polynomial $c\left(x\right)$ is given by $c\left(x\right)=3x^7+7x^6+x^5+3x^4+7x^3+4x^2+1Dx+1$, where the coefficients are represented in base-16 forms. A not given polynomial chosen polynomial is $m\left(x\right)=x^8+x^7+x^5+x^4+x+1$.

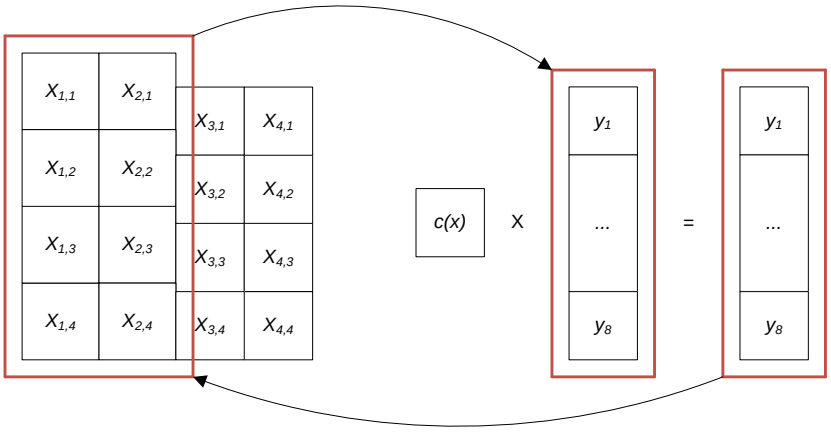

Fig. 2 Scheme of the execution of the $\mathrm{MixColumns}\left(state\right)$ operation

In $SubBytes_{Luna}\left(state\right)$ and $SubBytes_{\mathrm{Neptun}}\left(state, subkey\left[i\right]\right)$, an operation table is replaced according to each of the 16- and 8-bit data blocks with a specific table of substitutions (see Figs. 3a and 3b, respectively). Luna uses one 16×16 substitution table, while Neptune uses 16 tables, with the choice of a particular table in each round depending on the expanded key (a dynamically changeable substitutions table complicates the cryptanalysis and will dynamically manage the information

dispersion). The substitution table was constructed in such way that there were no fixed points, as well as to satisfy the respective equalities for the parameters, $\Delta = \Lambda = 2^{-14}$ for the Luna substitution table and $\Delta = \Lambda = 2^{-6}$ for each Neptune substitution table.

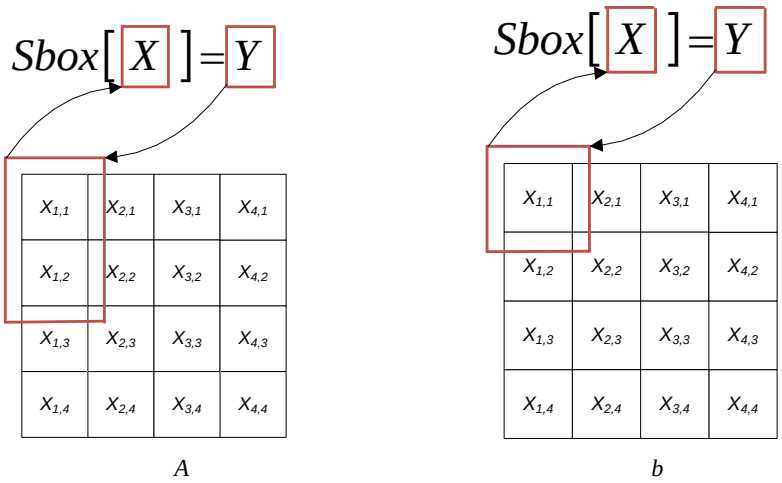

Fig. 3. Scheme of substitution table operation for (a) Luna and (b) Neptune

The proposed substitution tables are created by calculating the inverse field element $(C/X)^{-1} \in GF(2^q)$ and then performing affine transformations over the field $GF(2)$: $S(X) = M \cdot (C/X)^{-1} + V$, where $X, C, V \in GF(2^q)$, and M is not a singular square matrix over the $GF(2)$ field of $q \times q$ size. For Luna, $q = 16$, and for *Neptune*, $q = 8$. The C, V, and M settings for the Neptune and Luna table lookup algorithms are shown in base-16 forms in Tables 2 and 3, respectively (each matrix row is shown in the form of one base-16 number).

Table 2. C, V, and M settings for building the substitution table of the Luna encryption algorithm

M	*C*	*V*
{0652, CA4, 1948, 3290, 6520, CA40, 9481, 2903, 5206, A40C, 4819, 9032, 2065, 40CA, 8194, 0329}	1787	2544

Table 3. C, V, and M settings for building the substitution table of the Neptune encryption algorithm

Substitution table index	M	C	V
1	{ 91, 23, 46, 8C, 19,	95	E0
2	{ 83, 7, E, 1C, 38,	E1	E4
3	{ AB, 57, AE, 5D, BA,	14	EA
4	{ EA, D5, AB, 57, AE,	54	5
5	{ 94, 29, 52, A4, 49,	B2	B0
6	{ AB, 57, AE, 5D, BA,	50	7A
7	{ 64, C8, 91, 23, 46,	DD	F8
8	{ C4, 89, 13, 26, 4C,	F9	97
9	{ 2F, 5E, BC, 79, F2,	1D	B2
10	{ FE, FD, FB, F7, EF,	95	8E
11	{ D6, AD, 5B, B6, 6D,	13	43
12	{ A4, 49, 92, 25, 4A,	22	35
13	{ E9, D3, A7, 4F, 9E,	D9	B9
14	{ 7F, FE, FD, FB, F7,	EE	54
15	{ 8A, 15, 2A, 54, A8,	3B	EA
16	{ D3, A7, 4F, 9E, 3D,	91	E7

In $ShiftRows(state)$ operation executes byte matrix elements shift $state$: elements of i ($i = 2, 3$) last string sequence $state$ cyclically shifted right by 2 elements.

The procedures of Neptune and Luna decoding are similar to the encryption procedure (see Figs. 1a and 1b), except that the extended keys are provided in reverse order, with the reverse substitution tables and a reversed $\mathrm{MixColumns}(state)$ operation being used (multiplication by a polynomial $d(x) = 7Ax^7 + A1x^6 + F8x^5 + EEx^4 + 20x^3 + 89x^2 + EBx + 51$).

In the key expansion procedure, a $128n$-bit secret key K ($n = 1, 2, 4$) is divided into n parts of 128 bits (a_l, $l = \overline{1,n}$) to formulate extended keys. Each of these is divided into four k^l_i ($i = \overline{1,4}$) parts that, together with the 32-bit variables A, B, C, D, E, F, and y_i ($i = \overline{1,4}$), are moved to the key expansion routine input, the pseudocode of which is shown in Fig. 4.

The $Sbox(X)$ operation for Neptune and Luna performs tabular substitution in accordance with each 16 and 8 bits, respectively (Luna uses a substitution table based on the parameters from Table 2, while Neptune uses bases of the parameters from the first row of Table 3). $Mix(y_1, y_2, y_3, y_4)$ is the operation of linear dispersion. In this operation, variables y_i are broken into two parts of eight bytes, each of which is considered as a polynomial over the field $GF(2^8)$ with eight terms that are multiplied by $x^8 + 1$ modulo a fixed $c(x)$ polynomial of order seven, where

$c(x) = 3x^7 + 7x^6 + x^5 + 3x^4 + 7x^3 + 4x^2 + 1Dx + 1$. As a polynomial that is not chosen, consider $m(x) = x^8 + x^7 + x^5 + x^4 + x + 1$.

Input: a_l , $KolSubKey$

$A, B, C, D, E, F, y_1, y_2, y_3, y_4$.

Output: 128 -bit extended keys $subkey[i]$, $i = \overline{0, KolSubKey - 1}$

1. $For \quad k = 0, \quad k < KolSubKey, \quad k++ \quad do$

1.1. $subkey[k] = 0;$

2. $For \quad l = 0, \quad l < n, \quad l++ \quad do$

2.1. $For \quad k = 0, \quad k < KolSubKey, \quad k++ \quad do$

2.1.1. $For \quad j = 0, \quad j < 7, \quad j++ \quad do$

2.1.1.1 $A = \left(\left(A \oplus k^l_2\right) <<< \left(D \oplus k^l_1\right)\right) \oplus y_3;$

2.1.1.2. $k^l_1 = Sbox\left(\left(A \oplus k^l_1\right) + B\right);$

2.1.1.3. $B = Sbox\left(B \oplus k^l_1\right) >>> \left(C \oplus k^l_3\right);$

2.1.1.4. $k^l_2 = Sbox\left(\left(B \oplus k^l_2\right) <<< k^l_4\right) \oplus A;$

2.1.1.5. $y_1 = Sbox\left(\left(\left(Sbox\left(y_1 \oplus k^l_2\right) <<< k^l_1\right) \oplus E\right) \oplus y_4\right);$

2.1.1.6. $C = Sbox\left(\left(C \oplus F\right) + y_1\right) \oplus D;$

2.1.1.7. $y_2 = Sbox\left(\left(\left(y_2 \oplus C\right) >>> k^l_2\right) \oplus k^l_1\right);$

2.1.1.8. $Mix\left(y_1, y_2, y_3, y_4\right);$

2.1.1.9. $D = \left(\left(D \oplus k^l_4\right) <<< \left(A \oplus k^l_3\right)\right) \oplus y_1;$

2.1.1.10. $k^l_3 = Sbox\left(\left(D \oplus k^l_3\right) + E\right);$

2.1.1.11. $E = Sbox\left(E \oplus k^l_3\right) >>> \left(F \oplus k^l_1\right);$

2.1.1.12. $k^l_4 = Sbox\left(\left(E \oplus k^l_4\right) <<< k^l_2\right) \oplus D;$

2.1.1.13. $y_3 = Sbox\left(\left(\left(Sbox\left(y_3 \oplus k^l_4\right) <<< k^l_3\right) \oplus B\right) \oplus y_2\right);$

2.1.1.14. $F = Sbox\left(\left(F \oplus C\right) + y_3\right) \oplus A;$

2.1.1.15. $y_4 = Sbox\left(\left(\left(y_4 \oplus F\right) >>> k^l_4\right) \oplus k^l_3\right);$

2.1.1.16. $Mix\left(y_1, y_2, y_3, y_4\right);$

2.1.2 $temp[l][k] = y_1 \mid y_2 \mid y_3 \mid y_4;$

2.1.3 $subkey[k] = subkey[k] \oplus temp[l][k].$

Fig. 4. Pseudocode procedure for key expansion

Initial values of the variables A, B, C, D, E, F, y_i are listed in base-16 in Table 4.

Table 4. Initial values of variables A, B, C, D, E, F, y_i

Variable	Initial values
A	13C5E572
B	BC6FF4AD
C	4C1371E1
D	89F8D170
E	01069DA9
F	C5F52BD7
y_1	3B106B7A
y_2	7E15CEC1
y_3	23B0C13E
y_4	37A763D2

4. Software optimization of the Neptune and Luna cryptographic algorithms

Both Neptune and Luna utilize widely used algebraic operations in finite fields. Immediate execution of these operations would lead to extremely inefficient implementations. However, the byte-algorithm structure opens up opportunities for optimization. Thus, two-byte substitution can be implemented during Luna implementation, shifting and multiplying the result by the corresponding columns of the matrix M as one of substitution 16 bits on 64 bits, and Neptune implementation can be implemented as a bit-substitution shift, with a multiplication matrix *state* element on the column of the matrix M being implemented as 8-bit on 64-bit substitution. In that case, a complete round of Luna will consist of eight permutations of 16 on 64 bits and eight-bit addition modulo 2. Similarly, performing a full round of the Neptune algorithm requires only 16 substitutions of 8-bit on 64 bits and 16 additions modulo 2. Thus, by allocating a larger amount of RAM and performing preliminary calculations, it is possible to reduce the number of operations in the round and achieve a cryptographic processing speed boost.

5. Statistical Security Estimation using NIST STS

Properties of pseudorandom sequences formed with the help of Neptune and Luna (in counter mode) have been studied in the environment of NIST STS statistical tests (testing technique described in [17]). Statistical portraits of Neptune and Luna software implementations are shown in Figs. 5-6, respectively.

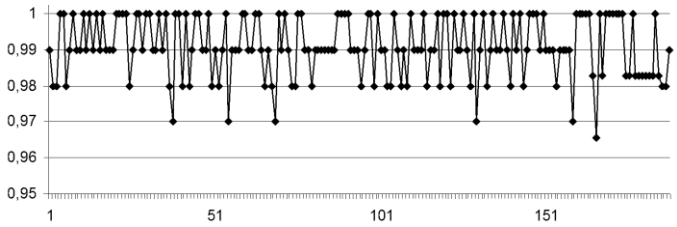

Fig. 5 Statistical portrait of Luna encryption algorithm in counter mode

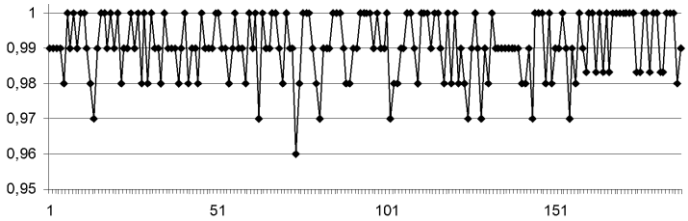

Fig. 6 Statistical portrait of Neptune encryption algorithm in counter mode

For comparison, Table 5 presents the results of testing sequences generated on the basis of the Luna, Neptune, Ukrainian Encryption Standard 28147-2009, Blum Blum Shub (BBS), and Kalyna algorithms. As can be seen from the results (Table 5), the Luna- and Neptune-based generators passed comprehensive testing using the NIST STS method and show better results than other algorithm-based generators.

Table 5. Results of sequence testing

Generator	Number of tests in which the test was conducted	
	99% sequence	96% sequence
BBS	133 (70,3%)	189 (100%)
DSTU 28147-2009	132 (69,8%)	188 (99,4%)
Kalyna	136 (72,0%)	189 (100%)
Luna	141 (74,6%)	189 (100%)
Neptune	140 (74,0%)	189 (100%)

6. Encryption Rate Estimation

Based on the described optimization software, the Luna, Neptune, AES, and Kalyna encryption algorithms and Ukrainian Encryption Standard 28147-2009 were implemented in the C++ programming language. During the Luna implementation, two-byte substitution, shifting, and multiplying the result by the corresponding columns of the matrix as a 16-bit to 64-bit substitution were presented. Similarly, during the Neptune and Kalyna implementations, byte operation substitution, shift, and multiplication of elements of the matrix by a column matrix as one eight-bit to 64-bit substitution were presented. During the DSTU 28147-2009 implementation, every two four-by-four-bit substitution tables were combined as a table of eight by eight bits, allowing reduction of the number of lookup operations from eight to four. During AES implementation, the operations of byte substitution, shift, and

multiplication of elements of the matrix by a column matrix as one eight-bit to 32-bit substitution were presented.

After software tools development, an experimental study was conducted to show that, in identical conditions, the Neptune and Luna encryption algorithms are 1.09 to 2.93 times faster than the Ukrainian Encryption Standard 28147-2009, Kalyna, or AES ciphers (see Table 6). The studies were conducted on an Intel(R) Core(TM)2 Duo T7300 2.0 GHz.

Table 6. Comparison of encryption algorithm speed performance

Encryption algorithm	Speed (MB/s)
AES -128	37.2
Kalyna -128	34.9
DSTU 28147-2009	18.1
Luna -128	53.1
Neptune -128	40.9

7. Security Estimation against Linear and Differential Cryptanalysis

During calculation of the analytical upper bounds of the parameters characterizing practical security to linear and differential cryptanalysis using formulas (1) and (2), Δ and Λ must be calculated depending on the parameters' lookup table. For this purpose, special software was developed and special tables were built using equations (3) and (4). Then, we determined the maximum value in these tables (except for the items in the zero-th row or column). As a result, it was determined that for Luna, $\Delta = \Lambda = 2^{-14}$, and for each Neptune table, $\Delta = \Lambda = 2^{-6}$.

Table 7 contains the analytical upper bounds of the parameters (using equations (1) - (2)), characterizing the practical security of the Neptune and Luna encryption algorithms to differential and linear cryptanalysis.

Table 7. Analytical upper bounds of the security against differential and linear cryptanalysis

Key length (bit)	*Luna*		*Neptune*	
	Differential cryptanalysis	Linear cryptanalysis	Differential cryptanalysis	Linear cryptanalysis
128	$EDP(\Omega) \le 2^{-392}$	$ELP(\Omega) \le 2^{-392}$	$EDP(\Omega) \le 2^{-222}$	$ELP(\Omega) \le 2^{-222}$
256	$EDP(\Omega) \le 2^{-512}$	$ELP(\Omega) \le 2^{-512}$	$EDP(\Omega) \le 2^{-330}$	$ELP(\Omega) \le 2^{-330}$
512	$EDP(\Omega) \le 2^{-770}$	$ELP(\Omega) \le 2^{-770}$	$EDP(\Omega) \le 2^{-546}$	$ELP(\Omega) \le 2^{-546}$

8. Conclusions

In this paper, two encryption algorithms were proposed to improve the efficiency of electronic information resources security from viewpoint of reliability, speed and security. As can be seen from the results of the experimental study, the proposed algorithms, Luna and Neptune, are at least two times faster than the previous Ukrainian National Encryption Standard 28147-2009 (in fact, this is the standard for all post-Soviet states). In addition, designed algorithms passed comprehensive control

using the NIST STS technique and showed better results than other encryption algorithm-based generators. It was also shown that the proposed algorithms are practically secured against linear and differential cryptanalysis.

References

1. Gnatyuk, S., Zhmurko, T., Falat, P.: Efficiency increasing method for quantum secure direct communication protocols. In: Proceedings of the 2015 IEEE 8th International Conference on "Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications" (IDAACS'2015), Warsaw, Poland, 468-472 (September 24-26, 2015).
2. Korchenko, O., Vasiliu, Ye., Gnatyuk, S.: Modern quantum technologies of information security against cyber-terrorist attacks. Aviation 14 (2), 58-69 (2010).
3. Kovtun, M., Kovtun, V., Okrimenko, A., Gnatyuk, S.: Search method development of birationally equivalent binary Edwards curves for binary Weierstrass curves from DSTU 4145-2002. In: Proceedings of 2nd International Scientific-Practical Conf. on the Problems of Info-communications. Science and Technology, Kharkiv, Ukraine, 135-139 (October 13-15, 2015).
4. Advanced Encryption Standard (AES): FIPS 197. Gaithersburg, Maryland, USA: NIST, 2001. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.
5. NESSIE contest (The New European Signature Algorithms, Integrity and Encryption) Panasenko, S. http://old.cio-world.ru/bsolutions/e-safety/340556.
6. Position about carrying out of open competition of cryptographic algorithms, Institute of Cybernetics named after V. Glushkov of NASU. http://www.dstszi.gov.ua/dstszi/control/ru/publish/article;jsessionid=EE63A37FEF8F5B34030F1E38D7247DBC?art_id=48387&cat_id=92733.
7. Gorbenko, I., Lisitskaya I.: Encryption algorithms standardization. Project requirements of national standard for block symmetric encryption on the modern stage of cryptography development. Radio Engineering, 5-10 (2011).
8. Gorbenko, I., Dolgov, V., Oliynykov, R. et al: Principles of construction and properties of IDEA-like block symmetric ciphers. Applied Radio Electronics 6 (2), 158-173 (2007).
9. DSTU 7624:2014. Ukrainian National Encryption Standard "Kalyna", 86 p. (2014).
10. Panasenko, S.: The encryption algorithms. Special guide, St. Petersburg, BHV-Petersburg. 576 (2009).
11. Alekseichuk, A., Kovalchuk, L., Skrynnik, E. et al: Rating of practical resistance of Kalyna block cipher relative to the difference methods, linear cryptanalysis and algebraic attacks based on homomorphisms. Applied Radio Electronics 7 (3), 203-209 (2008).
12. Alekseichuk, A., Kovalchuk, L., Skrynnik, E. et al: Rating of practical resistance of "Kalyna" block cipher relative to the difference methods, linear cryptanalysis and algebraic attacks based on cryptanalysis. In: Proceedings of the 4th intern. conf. on security and countering terrorism. MSU, M. V. Lomonosov. 30-31 Oct. 2008, 2th ed, M.: MCCME, 15-20 (2009).
13. Alsalami Y., Yeun C. Y., Martin T.: Linear and differential cryptanalysis of small-sized random (n, m)-S-boxes. In: Proceedings of 2016 11th International Conference for Internet Technology and Secured Transactions, Barcelona, Spain. DOI: 10.1109/ICITST.2016.7856751
14. Lai, X., Massey, J., Murphy, S.: Markov ciphers and differential cryptanalysis. Advances in Cryptology, EUROCRYPT'91, Proceedings, Springer Verlag, 17-38 (1991).
15. Matsui, M.: Linear cryptanalysis methods for DES cipher. EUROCRYPT, Springer Verlag (1998).
16. Daemen, J.: Cipher and hash function design strategies based on linear and differential cryptanalysis: Ph. D. Thesis, Katholieke Univ. Leuven (1995).
17. Narges, M., Khayyambashi, M. Performance Evaluation of Authentication Encryption and Confidentiality Block Cipher Modes of Operation on Digital Image. International Journal of Computer Network and Information Security (IJCNIS), Vol.9, No.9, pp.30-37, 2017. DOI: 10.5815/ijcnis.2017.09.04.