Cybernetics and Systems Analysis, Vol. 59, No. 1, January, 2023

KRIP: HIGH-SPEED HARDWARE-ORIENTED STREAM CIPHER BASED ON A NON-AUTONOMOUS NONLINEAR SHIFT REGISTER

L. V. Kovalchuk,[1] I. V. Koriakov,[2] and A. N. Alekseychuk[3] UDC 621.391:519.2:519.7

Absract. *A stream cipher based on a non-autonomous nonlinear shift register of length 2 over the alphabet of 2^{256} symbols is proposed. This register works similarly to the Feistel cipher with the round function used in the Kalyna encryption algorithm. It is shown that with the security level 2^{256}, the Krip cipher provides a fourfold gain in performance over the accepted standard of stream encryption in Ukraine and almost twentyfold when compared to the modern encryption algorithm Espresso.*

Keywords: *stream cipher, Feistel scheme, nonlinear shift register, generator of pseudorandom sequences, algebraic attacks, correlation attacks, Strumok, Espresso, Krip.*

INTRODUCTION

Recently, the use of stream ciphers has become increasingly popular due to the development of information technologies, data transmission, and significant progress in the study of cryptographic properties of stream encryption algorithms. Currently, stream ciphers are used in embedded applications of systems with a limited number of computing resources, in particular in wireless telephony and telecommunication switching systems and pay TV (for more detailed information see, e.g., [1]). The Strumok cipher [2], which is the national standard of stream encryption in Ukraine [3], takes a special place in this consideration.

Despite the variety of existing stream ciphers, the problem of creating hardware-oriented stream encryption algorithms that meet increased performance requirements and have acceptable circuit complexity remains relevant. As one of the possible approaches to solving the issue, this article proposes the stream encryption Krip algorithm based on a nonlinear shift register (NSR).

Unlike other stream ciphers based on NSR, such as Grain [4] and Espresso [5], the Krip algorithm uses a non-autonomous shift register of length 2 over the alphabet of cardinality 2^{256}, which functions similarly to the Feistel cipher with the round function used in the Kalyna encryption algorithm [6]. The problem of security level against a series of attacks for the proposed cipher thus could be reduced to the similar problem with the Feistel cipher.

We demonstrate that for security level at 2^{256} operations, Krip provides a four-fold speed gain over the Strumok algorithm and almost a twenty-fold gain over the Espresso algorithm, although it has a significantly higher hardware complexity than the latter.

[1]Institute of Physics and Technology of the National Technical University "Igor Sikorsky Kyiv Polytechnic Institute" and G. E. Pukhov Institute for Modelling in Energy Engineering, National Academy of Sciences of Ukraine, Kyiv, Ukraine, *lusi.kovalchuk@gmail.com*. [2]Krypton Research and Development Company Co., Ltd., Kyiv, Ukraine, *ikor@i.ua*. [3]National Technical University "Igor Sikorsky Kyiv Polytechnic Institute," Kyiv, Ukraine, *alex-dtn@ukr.net*.
Translated from Kibernetyka ta Systemnyi Analiz, No. 1, January–February, 2023, pp. 21–32. Original article submitted July 26, 2022.

The article has the following structure. Section 1 gives the definition of the Krip algorithm; Sec. 2 presents the results of the analysis of its security level against a number of known attacks; Sec. 3 presents the results of statistical testing of encrypted messages obtained using Krip; and Sec. 4 presents the characteristics of its hardware realization. A brief summary concludes the article.

1. DESCRIPTION OF THE ENCRYPTION ALGORITHM

For any natural n, let V_n be the set of binary vectors of the length n. For any integers i, j, put $\overline{i, j} = \{k \in \mathbf{Z}: i \le k \le j\}$.

The proposed algorithm includes two procedures:

1) encryption/decryption of messages using NSR;

2) formation of the initial state of the NSR with respect to the key $k \in V_{256}$ and the initialization vector $c \in V_{128}$.

Definition of an NSR. A nonlinear shift register is a finite automaton with the input alphabet $X = V_{256}$, output alphabet $Y = X$, the set of states $S = V_{512}$, and functions of transitions and outputs determined by the following formulas:

$$h((s_1, s_2), x) = (s_2 \oplus \varphi(x \oplus s_1), s_1), \tag{1}$$

$$f((s_1, s_2), x) = x \oplus \psi_1(s_1, s_2) \oplus \psi_2(s_1, s_2), \tag{2}$$

respectively, where $x \in X$, $s_1, s_2 \in V_{256}$, φ and $\psi = (\psi_1, \psi_2)$ are substitutions on the sets V_{256} and V_{512}, respectively, $\psi_i: V_{512} \to V_{256}$, $i \in \overline{1,2}$.

The described NSR works as usual: if the register is in the state (s_1, s_2), where $s_1, s_2 \in V_{256}$, and its input is the symbol $x \in X$, then the register produces the output symbol $y = f((s_1, s_2), x)$ and moves to the next state $h((s_1, s_2), x)$.

Encryption. Let $x_0, x_1, \ldots$ be a plaintext, where $x_i \in X$, $i = 0, 1, \ldots, (s_1, s_0)$, is the initial state of the NSR, $s_0, s_1 \in V_{256}$. Then the symbol y_i of the ciphertext at the ith cycle is determined by the formula

$$y_i = f((s_{i+1}, s_i), x_i), \tag{3}$$

where the sequence $s_0, s_1, \ldots$ is given by the recurrence relation $(s_{i+2}, s_{i+1}) = h((s_{i+1}, s_i), x_i)$, which can be presented as

$$s_{i+2} = s_i \oplus \varphi(x_i \oplus s_{i+1}), \ i = 0, 1, \ldots. \tag{4}$$

In more detail: let (s_1, s_0) be the initial state of the NSR and x_0 be the symbol of a plaintext at time $i = 0$. Then the ciphertext symbol in this cycle is calculated by the formula

$$y_0 = f((s_1, s_0), x_0) = x_0 \oplus \psi_1(s_1, s_0) \oplus \psi_2(s_1, s_0), \tag{5}$$

then the NSR goes to the next state $(s_2, s_1) = h((s_1, s_0), x_0)$, where $s_2 = s_1 \oplus \varphi(x_0 \oplus s_0)$.

Then, if the NSR input is a plaintext symbol x_1, then the output ciphertext symbol is

$$y_1 = f((s_2, s_1), x_1) = x_1 \oplus \psi_1(s_2, s_1) \oplus \psi_2(s_2, s_1) \tag{6}$$

and the register moves to the next state $(s_3, s_2) = h((s_2, s_1), x_1)$, where $s_3 = s_2 \oplus \varphi(x_1 \oplus s_1)$. Then the encryption continues in a similar way.

Decoding. Let the initial state (s_1, s_0) of the register and the symbol y_0 of the ciphertext at time $i = 0$ be known. Then, by formula (5), we can find the symbol of the plaintext x_0 and calculate the state $(s_2, s_1) = h((s_1, s_0), x_0)$ of the NSR at time $i = 1$. Next, using formula (6), by this state and the ciphertext symbol y_1, we can find the symbol x_1 and then calculate the state $(s_3, s_2) = h((s_2, s_1), x_1)$ of the NSR at time $i = 2$. In general, knowing the symbol y_i and the state (s_{i+1}, s_i) of the NSR at the ith cycle, we can calculate the symbol x_i by formula (3), and then find the state of the NSR at the $(i+1)$th cycle by the formula $(s_{i+2}, s_{i+1}) = h((s_{i+1}, s_i), x_i)$, $i = 0, 1, \ldots$.

$$M = \begin{pmatrix} x_{0,1} & x_{0,2} & x_{0,3} & x_{0,4} \\ x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} \\ x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} \\ x_{3,1} & x_{3,2} & x_{3,3} & x_{3,4} \\ x_{4,1} & x_{4,2} & x_{4,3} & x_{4,4} \\ x_{5,1} & x_{5,2} & x_{5,3} & x_{5,4} \\ x_{6,1} & x_{6,2} & x_{6,3} & x_{6,4} \\ x_{7,1} & x_{7,2} & x_{7,3} & x_{7,4} \end{pmatrix} \overset{L}{\mapsto} \begin{pmatrix} x_{0,1} & x_{0,2} & x_{0,3} & x_{0,4} \\ x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} \\ x_{2,2} & x_{2,3} & x_{2,4} & x_{2,1} \\ x_{3,2} & x_{3,3} & x_{3,4} & x_{3,1} \\ x_{4,3} & x_{4,4} & x_{4,1} & x_{4,2} \\ x_{5,3} & x_{5,4} & x_{5,1} & x_{5,2} \\ x_{6,4} & x_{6,1} & x_{6,2} & x_{6,3} \\ x_{7,4} & x_{7,1} & x_{7,2} & x_{7,3} \end{pmatrix}$$

Fig. 1. Transformation L.

Definition of the Substitution φ. Substitution φ is the round function used in the Kalyna cipher with a block length of 256 bits [6]. Note that this cipher uses four substitutions: $\pi_0, \pi_1, \pi_2, \pi_3$ on the set V_8, on which the structure of the field $\mathbf{GF}(2^8)$ is defined in a certain way, as well as the 8×8 matrix D over this field, which is maximally spatially separable [6, Secs. 5.3 and 5.5].

Denote $\sigma_i = \pi_{i \bmod 4}$, $i \in \overline{0,7}$.

Let $u \in V_{256}$; then, to calculate the value of $\varphi(u)$, the following algorithm is used:

(1) present the vector u in the form

$$u = (u_{0,1}, u_{1,1}, \ldots, u_{7,1}, u_{0,2}, u_{1,2}, \ldots, u_{7,2}, u_{0,3}, u_{1,3}, \ldots, u_{7,3}, u_{0,4}, u_{1,4}, \ldots, u_{7,4}),$$

where $u_{i,j} \in V_8$, and calculate the vector x with the coordinates $x_{i,j} = \sigma_i(u_{i,j})$, $i \in \overline{0,7}$, $j \in \overline{1,4}$, applying to each vector coordinate u the corresponding substitution from the set $\pi_0, \pi_1, \pi_2, \pi_3$;

(2) form the M matrix 8×4 from the vector x and apply the transformation L to it (Fig. 1);

(3) multiply each column of the matrix $L(M)$ on the matrix D over the field $\mathbf{GF}(2^8)$ and obtain the new 8×4 matrix with the elements $v_{i,j} \in V_8$, $i \in \overline{0,7}$, $j \in \overline{1,4}$. Presenting the columns of this matrix, starting from the first one, in the form of rows one by one, we obtain the vector

$$v = (v_{0,1}, v_{1,1}, \ldots, v_{7,1}, v_{0,2}, v_{1,2}, \ldots, v_{7,2}, v_{0,3}, v_{1,3}, \ldots, v_{7,3}, v_{0,4}, v_{1,4}, \ldots, v_{7,4}),$$

which is equal to the value of $\varphi(u)$.

Definition of the Substitution ψ. This substitution is constructed from the round function used in the Kalyna cipher with a block length of 512 bits.

More precisely: let $u \in V_{512}$; then to calculate the value of $\psi(u)$, the following algorithm is used:

(1) present the vector u in the form

$$u = (u_{0,1}, u_{1,1}, \ldots, u_{7,1}, u_{0,2}, u_{1,2}, \ldots, u_{7,2}, \ldots, u_{0,8}, u_{1,8}, \ldots, u_{7,8}), \tag{7}$$

where $u_{i,j} \in V_8$, and calculate the vector x with the coordinates $x_{i,j} = \sigma_i(u_{i,j})$, $i \in \overline{0,7}$, $j \in \overline{1,8}$, applying to each vector coordinate u the corresponding substitution from the set $\pi_0, \pi_1, \pi_2, \pi_3$;

(2) form the 8×8 matrix M' from the vector x and apply the transformation L' to it as shown in Fig. 2;

(3) multiply each column of the matrix $L'(M')$ on the matrix D over the field $\mathbf{GF}(2^8)$; as a result, we obtain the new 8×8 matrix with the elements $v_{i,j} \in V_8$, $i \in \overline{0,7}$, $j \in \overline{1,8}$. Present the columns of this matrix, starting from the first, in the form of rows one by one and obtain the vector

$$v = (v_{0,1}, v_{1,1}, \ldots, v_{7,1}, v_{0,2}, v_{1,2}, \ldots, v_{7,2}, \ldots, v_{0,8}, v_{1,8}, \ldots, v_{7,8});$$

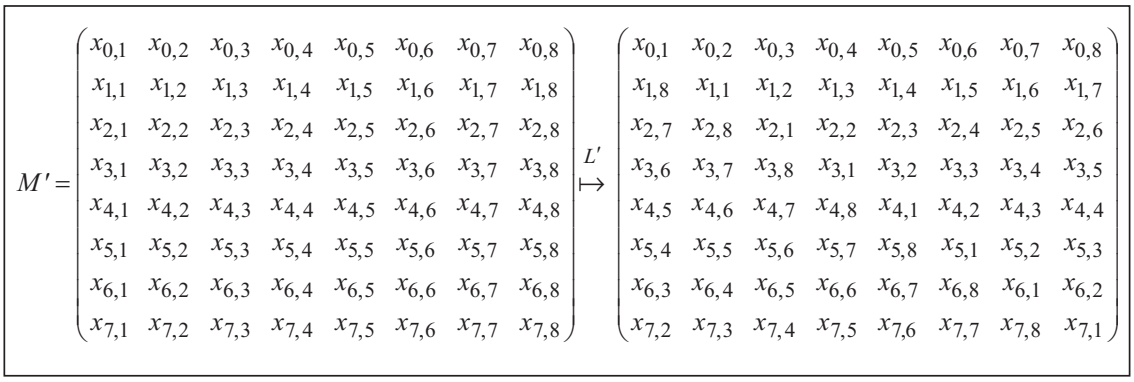

Fig. 2. Transformation L'.

TABLE 1. Values Used to Form the Initial State of the NSR

Constants	Constant Value
C_1	A09E667F 3BCC908B 2FB1366E A957D3E3 ADEC1751 2775099D A2F590B0 667322A9
C_2	B67AE858 4CAA73B2 5742D707 8B83B892 5D834CC5 3DA4798C 720A6486 E45A6E24
C_3	C6EF372F E94F82BE 73980C0B 9DB90682 1044ED7E 744E4A3F 0D8D423A 1831D2A4
C_4	54FF53A5 F1D36F1C EA7E61FC 37A20D54 A77FE7B7 8415DFC8 E34A6FE8 E2DF92A4
C_5	10E527FA DE682D1D E49E330E 42B4CBB2 9BA5A455 316E0C65 507CD18E 9E51E694
C_6	B05688C2 B3E6C1FD BD99E6FF 3C90BDC4 DBC64712 A5BB1687 67E27C3C F76C8E72

(4) applying to each vector coordinate v the corresponding substitution from the set $\pi_0, \pi_1, \pi_2, \pi_3$, determine the value of $\psi(u)$ as a vector with coordinates $\sigma_i(v_{i,j})$, $i \in \overline{0,7}$, $j \in \overline{1,8}$.

Determination of the Procedure for Formation of the Initial State of the NSR. The process of formation of the initial state of the NSR is determined on the basis of the scaled scheme of the Camellia block cipher [7]. The scaling is necessary because in the Camellia cipher, the block and key lengths are 128 and 256 bits, respectively, while in the Krip algorithm, the corresponding parameter values are 512 and 256 bits, respectively.

The constants C_1, …, C_6 used in the above procedure are determined by the following algorithm, similar to the algorithm of selecting the constants $\Sigma_{i(64)}$, $i \in \overline{1,6}$, in the Camellia key schedule [7, Sec. 3.4, Table 1].

The constants C_i, $i \in \overline{1,6}$, are computed as follows.

1. Calculate the value of $\sqrt{p_i}$, where p_i is the ith prime number in ascending order; the result is given in the binary number system with an accuracy of 261 binary digits after the decimal point. These 261 binary characters are written as a binary vector $c_i = (c_{i,1}, \ldots, c_{i,261})$.

2. Vector $C_i = (c_{i,5}, \ldots, c_{i,260})$ is derived from the vector c_i by subtracting the first four and the last coordinates.

3. The vector $C_i = (c_{i,5}, \ldots, c_{i,260})$ is divided into tetrads, each of which is converted into a corresponding symbol of the hexadecimal number system. The resulting vector consists of 64 hexadecimal characters and is the hexadecimal representation of the constant C_i. (Note that in the process of forming the initial state of the NSR, the appropriate binary representation of the constant C_i is used.)

The list of constants C_i in hexadecimal representation is given in Table 1.

The process of forming the initial state of the NSR by the key $k \in V_{256}$ and initialization vector $c \in V_{128}$ is as follows.

1. Write the vector in the register $(k_1 \oplus \bar{c}, k_1 \oplus k_2 \oplus c, k_1 \oplus k_2 \oplus \bar{c}, k_2 \oplus c)$, where $k = (k_1, k_2)$, $k_1, k_2 \in V_{128}$, and $\bar{c}$ denotes a vector with coordinates inverted to the coordinates of the vector c. Denote $K_L = (k_1 \oplus \bar{c}, k_1 \oplus k_2 \oplus c)$, $K_R = (k_1 \oplus k_2 \oplus \bar{c}, k_2 \oplus \bar{c})$.

2. Apply function (1) twice to the vector (K_L, K_R) using the constants C_1 and C_2; as a result, we obtain the vector $(U_1, U_2) = h(h((K_L, K_R), C_1), C_2)$.

3. Calculate the vector $(U_3, U_4) = (U_1, U_2) \oplus ((K_L, K_R) <<<_{128})$.

4. Apply function (1) twice to the vector (U_3, U_4) using the constants C_3 and C_4; the result is the vector $(K_{AL}, K_{AR}) = h(h((U_3, U_4), C_3), C_4)$.

5. Calculate $(U_5, U_6) = (K_{AL}, K_{AR}) \oplus ((U_3, U_4) <<<_{128})$.

6. Apply function (1) twice to the vector (U_5, U_6) using constants C_5 and C_6; the result is the vector $(K_{BL}, K_{BR}) = h(h((U_5, U_6), C_5), C_6)$.

7. Calculate the vectors $K_1, \ldots, K_{32}$ as follows:

$K_1 = K_L$, $K_{17} = K_{AL} <<<_{60}$,
$K_2 = K_R$, $K_{18} = K_{AR} <<<_{60}$,
$K_3 = K_{BL}$, $K_{19} = K_L <<<_{60}$,
$K_4 = K_{BR}$, $K_{20} = K_R <<<_{60}$,
$K_5 = K_{AL} <<<_{15}$, $K_{21} = K_{AL} \oplus (K_{BL} <<<_{77})$,
$K_6 = K_{AR} <<<_{15}$, $K_{22} = K_{AR} \oplus (K_{BR} <<<_{77})$,
$K_7 = K_L \oplus (K_{BL} <<<_{15})$, $K_{23} = K_{BL} <<<_{77}$,
$K_8 = K_R \oplus (K_{BR} <<<_{15})$, $K_{24} = K_{BR} <<<_{77}$,
$K_9 = K_{AL} <<<_{30}$, $K_{25} = K_{AL} <<<_{94}$,
$K_{10} = K_{AR} <<<_{30}$, $K_{26} = K_{AR} <<<_{94}$,
$K_{11} = K_{BL} <<<_{30}$, $K_{27} = K_L \oplus (K_R <<<_{94})$,
$K_{12} = K_{BR} <<<_{30}$, $K_{28} = K_R \oplus (K_{AL} <<<_{94})$,
$K_{13} = K_L \oplus (K_{AL} <<<_{45})$, $K_{29} = K_{AL} \oplus (K_{AR} <<<_{111})$,
$K_{14} = K_R \oplus (K_{AR} <<<_{45})$, $K_{30} = K_{AR} \oplus (K_{AL} <<<_{111})$,
$K_{15} = K_{BL} <<<_{45}$, $K_{31} = K_{BL} \oplus (K_{BR} <<<_{111})$,
$K_{16} = K_{BR} <<<_{45}$, $K_{32} = K_{BR} \oplus (K_{BL} <<<_{111})$.

8. Calculate the vector $(K_1^*, K_2^*) = h(h \ldots h(h((K_L, K_R), K_1), K_2) \ldots, K_{32})$ applying function (1) to the input vector (K_L, K_R) 32 times and using vectors $K_1, \ldots, K_{32}$ as input symbols calculated in item 7. The resulting vector (K_1^*, K_2^*) is the output of the procedure.

2. SECURITY LEVEL ANALYSIS

Algebraic Attacks. Let the plaintext $x_0, x_1, \ldots$ and its corresponding encrypted text $y_0, y_1, \ldots$ obtained by the initial state (s_1, s_0) of the NSR and formulas (3), (4) be known. Then, based on formulas (1), (2), to restore the initial state, we can write the system of equations

$$\psi_1(s_{i+1}, s_i) \oplus \psi_2(s_{i+1}, s_i) = x_i \oplus y_i, \tag{8}$$

$$(s_{i+1}, s_i) = (h_{x_i} \circ \cdots \circ h_{x_0})((s_1, s_0)), \ i = 0, 1, \ldots,$$

where the symbol $\circ$ denotes the operation of composition of mappings, $h_x(s', s'') = h((s', s''), x)$ for any $s', s'' \in V_{256}$ and $x \in X$.

System (8) contains 512 variables, i.e., vector coordinates (s_1, s_0); however, it can be replaced by an equivalent system of equations with 256 variables. For this, we should use the definition of the substitution ψ and the fact that arbitrary equality of the form $\psi_1(u) \oplus \psi_2(u) = w$ is equivalent to the equality that expresses certain 256 coordinates of the vector u in its representation (6) by the other 256 coordinates of this vector (this expression is given by a nonlinear function that depends on the substitutions $\pi_0, \pi_1, \pi_2, \pi_3$, the matrix D, and transformation L' presented in Fig. 2).

Note that the algebraic immunity of each substitution $\pi_0, \pi_1, \pi_2, \pi_3$ is equal to 3 [8]. Attempts to use conventional methods for solving such system that are more effective than the full search, result in fundamental difficulties associated with the analytical complexity of the system of equations and the unpredictability of their numerical characteristics (in particular, algebraic degree). Therefore, the security level of the Krip algorithm against algebraic attacks is currently estimated at the level of 2^{256}.

Attacks Based on Homomorphisms. If there is a homomorphism of an NSR into an automaton with the same input and output alphabets with a smaller number of states, then attacks based on homomorphisms can be applied to the encryption algorithm above [9, 10]. A sufficient condition for the inability of using such attacks is the primitivity of the group generated by the substitutions h_x, where $x \in X$, and the function h is determined by formula (1). In [11], a criterion for the primitivity of the group of substitutions of the specified form was obtained that implies that this group is primitive if the group generated by the substitution φ and all substitutions of the form $x \mapsto x \oplus k$, where $x, k \in V_{256}$, is primitive. However, according to Statement 4 from [12], the last group is sign-changing on the set V_{256}, and therefore primitive, which guarantees the security of the Krip algorithm against attacks based on homomorphisms.

Correlation Attacks. Since Krip does not contain linear components, classical correlation attacks on filter or combinatorial gamma generators based on linear shift registers [13] are irrelevant. From the system of equations (8), it follows that the problem of finding linear approximations of transformations implemented using the Krip algorithm and the problem of constructing linear attacks on a Feistel cipher with a round function φ, which is used in the Kalyna encryption algorithm with a block length of 256 bits, are closely related. Currently, there are no known similar (effective) attacks.

Note also that using Corollary 2 from [14] and the data given in Table 2 from [12], it is easy to see that the nonlinearity of the function $F(s) = \psi_1(s) \oplus \psi_2(s)$, $s \in V_{512}$ determined by the formula $\mathrm{nl}(F) = 2^{511} - 1/2 \cdot \max\limits_{\substack{v \in V_{256} \setminus \{0\}, \\ u \in V_{512}}} \left| \sum\limits_{s \in V_{512}} (-1)^{vF(s) \oplus us} \right|$ is not less than $2^{511}(1 - 9^4 \cdot 2^{-32})$.

Structural properties of internal sequences of the NSR. Consider the sequence of states $s_0, s_1, \ldots, s_t$ that is created by the initial state $s_0 \in S$ and the input sequence $x_0, x_1, \ldots, x_{t-1}$ of the register according to the rule $s_{i+1} = h(s_i, x_i)$, $i \in \overline{0, t-1}$.

Stetement 1. Let the element s_0 and the sequence $x_0, x_1, \ldots, x_{t-1}$ are chosen independently randomly with equal probability from the sets S and X^t, respectively. Then the probability that all the states $s_0, s_1, \ldots, s_t$ are pairwise different, is bounded from below by the value $1 - \dfrac{t(t-1)}{2|S|}$.

Proof. Consider a directed graph on the set of vertices S, such that from the vertex s to the vertex $\widetilde{s}$ an arc marked with a symbol $x \in X$ is directed, if and only if $h(s, x) = \widetilde{s}$. It is easy to see that for arbitrary elements $s, \widetilde{s} \in S$ and any natural $l \geq 2$, there exists exactly $|X|^{l-2}$ paths in this graph, directed from the vertex s to the vertex $\widetilde{s}$. Whence it follows that for any integers $0 \leq i < j \leq t-1$ and arbitrary $s = (s', s'') \in S$, the probability of the event $\{s_i = s_j = s\}$ is determined by the following equality:

$$\mathbf{P}\{s_i = s_j = s\} = \begin{cases} |X|^{-4} & \text{if } j - i \geq 2; \\ |X|^{-3} & \text{if } j - i = 1 \text{ and } s' = s''; \\ 0 & \text{otherwise.} \end{cases}$$

Hence, the probability that a random sequence $s_0, s_1, \ldots, s_t$ contains at least two identical elements, is equal to

$$\mathbf{P}\left(\bigcup_{\substack{0 \leq i < j \leq t-1, \\ s \in S}} \{s_i = s_j = s\} \right) \leq \sum_{\substack{0 \leq i < j \leq t-1: \, j-i \geq 2, \\ s \in S}} \mathbf{P}\{s_i = s_j = s\} + \sum_{\substack{0 \leq i < j \leq t-1: \, j-i = 1, \\ s \in S}} \mathbf{P}\{s_i = s_j = s\}$$

$$= \sum_{0 \leq i < j \leq t-1: \, j-i \geq 2} |S| \cdot |X|^{-4} + \sum_{0 \leq i < j \leq t-1: \, j-i = 1} |X| \cdot |X|^{-3} \leq \frac{t(t-1)}{2|X|^2},$$

which completes the prove.

This statement proves that the so-called birthday paradox holds for the sequence of states of the NSR, which is obtained from a randomly chosen (independent and equiprobable) initial state and input sequence, as well as for a purely random sequence of elements of the set S. In particular, with a high probability, there are no repetitions in the sequence of NSR's states if t is significantly smaller than $\sqrt{|S|} = 2^{256}$. Therefore, the relative share of the pairs (initial state, input sequence of the length t), for which the sequence of states contains a repeating fragment, is negligibly small if t is significantly smaller than 2^{256}.

3. RESULTS OF STATISTICAL ANALYSIS

Statistical analysis of the encryption algorithm carried out in the following directions.

1. Verification of the statistical quality of the algorithm as a generator of pseudorandom sequences.

2. Checking the independence of internal sequences obtained for the same key, different initialization vectors, and the null sequence as a plaintext.

Some details of this analysis are provided below.

Verification of the Statistical Quality of the Algorithm as a Generator of Pseudorandom Sequences. The verification was performed according to the methodology described in [15]: each of the 16 tests of the NIST package is applied with a level of significance $\alpha = 0.01$ to n encrypted messages of the length 10^6 bit received for open zero messages for the key (randomly generated)

3C4920A999A32DFE892761617E5FED65E5B5F5E41E7D97FC4993E05EFC8BA06B

and 1000 different random initialization vectors, where $n = 6233$ for the RandomExcursion and RandomExcursionVariant tests (this number of sequences is chosen for certain technical reasons related to the specifics of these tests) and $n = 10000$ for all other tests.

The following operations were performed for each test:

(1) n P-values $P_1, \ldots, P_n$ are obtained (the definition of P-value is given in [15]);

(2) the value of $\chi^2 = \sum_{i=1}^{10} \frac{(F_i - n/10)^2}{n/10}$, where n is the number of sequences in the statistical experiment and F_i is the number of P-values belonging to the interval $\left(\frac{i-1}{10}, \frac{i}{10}\right)$, $i \in \overline{1,10}$, is calculated;

(3) the value $P_T = \text{igams}(9/2,\ \chi^2/2)$, where igams is the incomplete gamma function defined by the formula

$$\text{igams}(a, x) = \frac{1}{\Gamma(a)} \int_x^\infty t^{a-1} e^{-t}\, dt, \text{ where } \Gamma(z) = \int_0^\infty t^{z-1} e^{-t}\, dt,$$

is calculated.

According to [15], the generator under consideration passes the test if the following two conditions are satisfied:

$$1 - 0.01 - 3\sqrt{\frac{0.01(1-0.01)}{n}} \le \frac{k}{n} \le 1 - 0.01 + 3\sqrt{\frac{0.01(1-0.01)}{n}}, \tag{9}$$

where k is the number of sequences that passed the test; and

$$P_T \ge 0.0001. \tag{10}$$

For the Krip encryption algorithm, conditions (9) and (10) are satisfied for all 16 tests. For the RandomExcursion and RandomExcursionVariant tests, formula (9) determines the interval (0.98622, 0.99378); and the interval (0.98702, 0.99298) for all other tests. The value of proportion k/n for the tests falls into the specified intervals (for the RandomExcursion and RandomExcursionVariant tests it coincides with the left boundary of the interval). Therefore, according to [15], Krip can be considered a generator of pseudorandom sequences.

TABLE 2. Values of the Determinant of the Correlation Matrix Based on 20 experiments

Key Number	Key Value	Value of the Determinant
1	3C4920A999A32DFE892761617E5FED65 E5B5F5E41E7D97FC4993E05EFC8BA06B	0.995105
2	E87D2A7AB61AE3C1D110E2411DDBBBC4 5C8D1564907631DBC454B6BC336E6EFC	0.995111
3	BD16E669884A36BF40E1BA5D9A7966AC A9D51B6EF44D3861A3EC93396AF829A0	0.995235
4	3F15DEF66809AC3551BEDC15BF1616C9 8FCDBFD881E7A5FB17867A73AE26EB3	0.995101
5	57F5C606C18C91B028496B4C8832875E B8C8937D1EF36C29CBB624B13CE7B5CC	0.995236
6	B0D4F772B4D8668266DA794C3F187A7D CB1BBDA921026AAE20E6F5E885B51BEC	0.995081
7	30A8DBDE96874904CA3E238D38ED2FC5 91CB7124D68925CFF6E9E75782E5CA56	0.995031
8	D031E8881C34BDD6E63C68BCFF0E96FC F7F8F7E56BCDCB5A5630AA287F2CD8DF	0.995208
9	AE2E3E05DB0D5EC0FEC779C46DDD6F30 95BF47B274FCDE67D508EBB0675752B	0.995162
10	43A5F2C4B6774DA5EDD3CE4DC47AAB6B 6893ED09B8D05B557DEAD91DD1B1C210	0.995138
11	7B533721E7615EFFE6D04EC528232A8E 705E5591EA286DA95312E7E57B311F7	0.995102
12	D816A9527C96981CE33E2681FF0855B BFB329EB282FCEE93F8F396171A5660E	0.995204
13	84E3C5A5B8F5B20385B1FB5556AA1880 675482745E7DE9C73A92CC45A78420C	0.995168
14	9EA55C5369DA43956B8897B0136DFEE6 86FDC7856220199EF9E99A0B6566D32	0.995280
15	40B2598A529AB9DC9C84D3222572D1CA 9BE02624299B29B99F2431ABB12EC3EB	0.995141
16	A257D64FCA7D62549EAFAB8B6BAA3F54 CA766D66BA2BFA8FC5B7964D42191D25	0.995123
17	EAA1B8B45F171CB6FED976239DB91194 598B4519B682298DEABC6BBA1F2F5D28	0.995262
18	E99A9E14E8B4710BB852708118303EA 6A828539830EC4E082D98D6787E7D3F7	0.995168
19	20CEE16EE94997247064103DBFC9FB6 32E321F5B578269EF2069E41D1B36A5	0.995093
20	E4A1CC943D0C6D7EF191473274E343C C2FFC014A95D8003FD4FC8BB1B7FC8D	0.995058
Sample mathematical expectation		0.99515035

Verification of the Independence of Internal Sequences of NSR. To perform this statistical analysis, the results of [16] were applied to verify the pairwise independence of an arbitrary number of random sequences using the sample correlation matrix.

Twenty experiments were performed as a part of statistical analysis. Each experiment included formation of $m = 100$ sequences with the length $l = 10^6$ bit each for a randomly generated key. Each sequence in turn was constructed from the zero input sequence and 100 randomly selected initialization vectors $IV_1, \ldots, IV_{100}$.

Each experiment consists of the following steps:

(1) form the bit sequence $\Xi^{(j)} = (\xi_1^{(j)}, \ldots, \xi_l^{(j)})$ obtained from the internal sequence corresponding to the vector IV_j, $j \in \overline{1, 100}$;

TABLE 3. The Value of the Determinant of the Correlation Matrix for 20 Series of 100 Sequences of the Length of 1,000,000 Bits Generated by a Certified Physical Random Sequence Generator

Series Number	Value of the Determinant	Series Number	Value of the Determinant
1	0.995169	11	0.995223
2	0.995150	12	0.995219
3	0.995164	13	0.995193
4	0.995178	14	0.995174
5	0.995172	15	0.995205
6	0.995182	16	0.995188
7	0.995163	17	0.995219
8	0.995168	18	0.995247
9	0.995166	19	0.995227
10	0.995201	20	0.995205
Sample mathematical expectation			0.99519065

(2) use the obtained sequences to determine the values

$$D_{ij} = \frac{1}{l-1}\sum_{k=1}^{l}(\xi_k^{(i)} - 0.5)(\xi_k^{(j)} - 0.5),$$

which are sample pairwise covariances, and construct the sample correlation matrix

$$\mathbf{R} = (R_{ij})_{i,\,j=1}^{100}, \text{ where } R_{ij} = \frac{D_{ij}}{\sqrt{D_{ii}}\sqrt{D_{jj}}}, \; i, j \in \overline{1, 100}.$$

Note that for the pairwise independent random sequences $\Xi^{(j)}$, $j \in \overline{1, 100}$, the matrix $\mathbf{R}$ coincides with the identity matrix I, and hence, its determinant is equal to one. However, in our case, this matrix is defined as an empirical one as it results from statistical experiments. Therefore, to construct a hypothesis testing criterion H_0 that "the sequences are pairwise independent" with the significance level $\alpha_0 = 10^{-3}$, we use the determinant of the matrix $\mathbf{R}$: $d = \det \mathbf{R}$.

The law of the determinant d distribution is rather complicated, but for sufficiently large values l, its asymptotic representation can be used: if the sequences are pairwise independent, then

$$\mathbf{P}\{-nd \le v\} = \mathbf{P}\{\chi_f^2 \le v\} + \frac{\gamma}{n^2}(\mathbf{P}\{\chi_{f+4}^2 \le v\} - \mathbf{P}\{\chi_f^2 \le v\}) + \mathrm{O}(n^{-3}),$$

where $f = \frac{m(m-1)}{2}$, $n = l - \frac{2m+11}{6}$, $\gamma = \frac{m(m-1)(2m^2-2m-13)}{288}$, and χ_f^2 is a random variable that has χ-squared distribution with f degrees of freedom.

Note that for $l = 10^6$ and $m = 100$ the following approximation

$$\mathbf{P}\{-nd \le v\} \approx \mathbf{P}\{\chi_f^2 \le v\} \tag{11}$$

can be used. Since positive values in the right-hand side of (11) are removed in the case of approximation, the obtained critical region corresponds to a higher significance level than the given one.

The critical region for the parameter $d = \det R$ with the significance level of $\alpha_0 = 10^{-3}$ belongs to $d \in (0, e^{-1})$ according to (11).

Thus, the values of d (see Table 2) provide provide the empirical evidence that supports the hypothesis H_0 on the pairwise independence of internal sequences of NSR. For comparison, Table 3 presents the values of d for 20 series from 100 sequences with 10^6 random bits in each.

TABLE 4. Characteristics of the Hardware Realizations of Strumok, Krip, and Espresso Algorithms

Parameter	Parameter Values for the Algorithms		
	Espresso	Strumok	Krip
Word length, bit	1	64	256
Clock frequency, MHz	2217	350	350
Number of conversion cycles	1	2	2
Number of logical blocks	1497	3100	8015
Number of 256×64 memory blocks	–	64	768
Number of 256×8 memory blocks	–	8	64
Encryption speed, Gbit/sec	2.22	11.2	44.8

4. COMPARISON OF CHARACTERISTICS OF HARDWARE REALIZATIONS OF STRUMOK, KRIP, AND ESPRESSO STREAM ENCRYPTION ALGORITHMS

Table 4 shows the characteristics of the projects of hardware realizations of the Strumok and Krip algorithms in the Quartus Prime 17.1 Lite Edition design software, as well as the corresponding characteristics of the hardware realization of the Espresso cipher.

Note that for the minimum possible number of the cycles (in this case 2), the hardware realization of the Strumok algorithm encrypts words with the length of 64 bits, and the realization of the Espresso algorithm with the length of 1 bit; at the same time, the realization of the Krip algorithm encrypts words with a length of 256 bits. Thus, we obtain a four-fold and a twenty-fold gain in speed for the Krip algorithm, respectively. Moreover, it should be noted that this gain involves much higher hardware costs. In particular, the Strumok algorithm implements a 64-bit element of the round function of the Kalyna algorithm, while the Krip algorithm contains two round functions used in the versions of Kalyna with block lengths of 256 and 512 bits, respectively, and the Espresso algorithm does not use table elements at all.

The main resource for increasing the performance when implementing these two round functions is the memory needed to store the tables. Modern FPGA integrated circuits contain a significant number of memory blocks sufficient to implement the Krip algorithm at a moderate hardware cost. For example, using Intel's Arria V GT 5AGTC3 FPGA with 1051 memory blocks and four 10.3125 Gbit/sec transceivers enables data encryption in networks at speeds greater than 40 Gbit/sec.

CONCLUSIONS

The article proposes a hardware-oriented Krip stream encryption algorithm. This algorithm is constructed based on a non-autonomous nonlinear shift register of length 2 over the alphabet of cardinality 2^{256} and is a high-speed stream cipher with a level of security that meets modern requirements. In the future, we plan to implement the Krip algorithm into a series of high-speed encryption tools for cryptographic protection of information circulating on the Ethernet network between Data Processing Centers. Such hardware is expected to provide the line-speed encryption (i.e., no packet loss) with low introduced delays not only in point–point topologies, but also in other multipoint ones.

The realization of the Krip algorithm allows us to perform data encryption at the L2 level according to the ISO model in metropolitan area network (Metro Ethernet) and in the carrier network (Carrier Ethernet) without degrading the parameters of high-speed optical communication channels built on Cisco NexusN5K-C5672UP equipment (about six 40G QSFP).

REFERENCES

1. A. Yu. Storozhuk, Evaluation and Provability Methods of the Stream Ciphers' Security against Statistical Attacks Based on Algebraic Degenerate Approximations of Boolean Functions [in Ukrainian], Ph.D. Thesis, National Aviation University, Kyiv (2016).
2. I. Gorbenko, A. Kuznetsov, Yu. Gorbenko, A. Alekseychuk, and V. Timchenko, "Strumok keystream generator," in: 2018 IEEE 9th Intern. Conf. on Dependable Systems, Services and Technologies (DESSERT) (Kyiv, Ukraine, May 24–27, 2018), IEEE (2018), pp. 294–299. https://doi.org/10.1109/DESSERT.2018.8409147.
3. The National Standard of Ukraine "DSTU 8845:2019, Information technologies, Cryptographic data security, Symmetric block transformation algorithm," DP "UkrNDNTs," Kyiv (2019).
4. M. Hell, T. Johansson, A. Maximov, and W. Meier, "The Grain family of stream ciphers," in: M. Robshaw and O. Billet (eds.), New Stream Cipher Designs; Lecture Notes in Computer Science, Vol. 4986, Springer, Berlin–Heidelberg (2008), pp. 179–190. https://doi.org/10.1007/978-3-540-68351-3_14.
5. E. Dubrova and M. Hell, "Espresso: A stream cipher for 5G wireless communication systems," Cryptology ePrint Archive: Report 2015/241 (2015). URL: http://eprint.iacr.org/2015/241.
6. R. Oliynykov, I. Gorbenko, O. Kazymyrov, V. Ruzhentsev, O. Kuznetsov, Yu. Gorbenko, O. Dyrda, V. Dolgov, A. Pushkaryov, R. Mordvinov, and D. Kaidalov, "A new encryption standard of Ukraine: The Kalyna Block Cipher," Cryptology ePrint Archive: Report 2015/650. URL: http://eprint.iacr.org/2015/650.
7. A. Aoki, T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakajima, and T. Tokita, "Camellia: A 128-bit block cipher suitable for multiple platforms — Design and analysis," in: Selected Areas in Cryptography, 7th Annual Intern. Workshop, SAC 2000 (Waterloo, Ontario, Canada, August 14–15, 2000), Proceedings; Lecture Notes in Computer Science, Vol. 2012, Springer (2001), pp. 39–56.
8. R. Oliynykov, I. Gorbenko, O. Kazymyrov, V. Ruzhentsev, and Iu. Gorbenko, "Design principles and main properties of the new Ukrainian national standard of block encryption," Ukr. Inf. Secur. Res. J., Vol. 17, No. 2, 142–157 (2015).
9. I. G. Shaposhnikov, "Congruences of finite multibase universal algebras," Diskr. Mat., Vol. 11, Iss. 3, 48–62 (1999).
10. Yu. N. Gorchinskii, "Homomorphisms of multibase universal algebras in the context of cryptographic applications," in: Trans. Discrete Mathematics, Vol. 1, TNP, Moscow (1997), pp. 67–84.
11. A. N. Alekseychuk and L. V. Skrypnik, "The primitivity criterion of the permutation group generated by the encryption round functions of Feistel cipher," Radiotekhnika, Iss. 141, 31–39 (2005).
12. A. N. Alekseychuk, L. V. Kovalchuk, A. S. Shevtsov, and S. V. Yakovliev, "Cryptographic properties of a new National encryption standard of Ukraine," Cybern. Syst. Analysis, Vol. 52, No. 3, 351–364 (2016). https://doi.org/10.1007/s10559-016-9835-0.
13. W. Meier, "Fast Correlation Attacks: Methods and Countermeasures," in: A. Joux (eds.), Fast Software Encryption, FSE 2011; Lecture Notes in Computer Science, Vol. 6733, Springer, Berlin–Heidelberg (2011), pp. 55–67. https://doi.org/10.1007/978-3-642-21702-9_4.
14. S. Park, J. Sung, S. Lee, and J. Lim, "Improving the upper bound on the maximum differential and the maximum linear hull probability for the SPN structures and AES," in: T. Johansson (eds.), Fast Software Encryption, FSE 2003; Lecture Notes in Computer Science, Vol. 2887, Springer, Berlin–Heidelberg (2003), pp. 247–260. https://doi.org/10.1007/978-3-540-39887-5_19.
15. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications: NIST Special Publication 800-22, Rev. 1 (1999).
16. T. W. Anderson, An Introduction to Multivariate Statistical Analysis, 3rd ed., Wiley (2003).