ISSN 2519-2310

COMPUTER SCIENCE AND CYBERSECURITY

ISSUE 1(9) 2018

V. N. Karazin Kharkiv National University Publishing

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ
MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені В.Н.КАРАЗІНА
ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ имени В.Н. КАРАЗИНА
V.N. KARAZIN KHARKIV NATIONAL UNIVERSITY

КОМП'ЮТЕРНІ НАУКИ ТА КІБЕРБЕЗПЕКА
КОМПЬЮТЕРНЫЕ НАУКИ И КИБЕРБЕЗОПАСНОСТЬ
COMPUTER SCIENCE AND CYBERSECURITY
(CS&CS)

Issue 1(9) 2018

Заснований 2015 року

Міжнародний електронний науково-теоретичний журнал
Международный электронный научно-теоретический журнал
International electronic scientific journal

The journal publishes research articles on theoretical, scientific and technical problems of effective facilities development for computer information communication systems and on information security problems based on advanced mathematical methods, information technologies and technical means.

Journal is published quarterly.
Approved for placement on the Internet by Academic Council of the Karazin Kharkiv National University (Nov. 26, 2018, protocol No.12)

Editorial office:
Karazin Kharkiv National University
Svobody Sq., 6, office 315a, Kharkiv, 61022, Ukraine (*North building of University, 3th floor*)
Phone: +38 (057) 705-10-83
E-mail: *cscsjournal@karazin.ua*
Web-page: *http://periodicals.karazin.ua/cscs (Open Journal System)*

Published articles have been internally and externally peer reviewed

TABLE OF CONTENTS

Issue 1(9) 2018

UDC 004.056.55

"STRUMOK" STREAM CIPHER

Ivan Gorbenko[1,2], Alexandr Kuznetsov[1,2], Yuriy Gorbenko[2], Anton Alekseychuk[2,3], Vlad Tymchenko[1]

[1] V. N. Karazin Kharkiv National University, 4 Svobody Sq., Kharkiv, 61022, Ukraine
[2] JSC "Institute of Information Technologies", 12 Bakulin St., Kharkiv, 61166, Ukraine
[3] National Technical University of Ukraine "Kyiv Polytechnic Institute", 37 Prospect Peremogy, Kiev 03056, Ukraine

gorbenkoi@iit.kharkov.ua, kuznetsov@karazin.ua, gorbenkou@iit.kharkov.ua, alex-dtn@ukr.net, tvlad.tyma@gmail.com

Reviewer: Nikołay Karpinskiy, Doctor of Sciences (Eng.), Full Prof., University of Bielsko-Biala, Willowa 2, Bielsko-Biala, 43-309, Poland.
mkarpinski@ath.bielsko.pl

Received on February 2018

Анотація: *This work presents the main developing results of a new keystream generator, which named "Strumok", and offered as a candidate for the national symmetric encryption standard of Ukraine. "Strumok" is built on SNOW 2.0-likes schema of the summation generator. Increased secret key length and the initialization vector allow using reliably the stream cipher even taking into account quantum cryptographic analysis methods. Unlike SNOW 2.0, Strumok is designed for use in more powerful 64-bit computing systems. The conducted comparative tests have shown that the "Strumok" on 32-bit computing systems also shows good performance results. There are basic transformation and individual results from the cipher performance research, here is it is shown the generator, which is capable of forming a keystream at speeds exceed of 10 Gbit per sec.*

Keywords: *encryption; stream cipher; synchronous keystream generator; pseudorandom sequence.*

1 Introduction

An important cryptographic information protection mechanism is a stream symmetric encryption [1,2]. It uses for providing services of information confidentiality and integrity (*as an additional service*) during data processing in information, telecommunication and information-telecommunication systems [1].

Recent years the requirements for modern streaming encryption algorithms have significantly increased [3-5]: on the one side, it is necessary to provide a high-speed cryptographic transformation (*more than 10 Gbit per sec*), on the other side, it have to withstand effectively for the latest methods of cryptographic analysis including methods using of quantum computation. Therefore, the development, research and a gradual introduction of new stream symmetric encryption methods is an actual and extremely important the national level scientific and applied problem. Usually a streaming encryption operation is a bitwise XOR operation between a keystream and a message. ISO/IEC 18033-4:2011 describes the output functions for different stream ciphers and certain pseudorandom numbers generators, which used for restricted information protecting, in particular to ensure the information confidentially when processing is going on [6]. The work objective is to present the main results of new pseudorandom number generator (a keystream) developing, which named "Strumok", and is proposed as a candidate for the national symmetric encryption standard of Ukraine [7-18]. The "Strumok" generator provides a high forming keystream rates (*over 10 Gbit per sec*) that exceeds most known algorithms and is suitable for using in a post-quantum environment.

2 General Parameters

In basics of Strumok algorithm lies the classical summing generator schema [1,2,6,7], which similar to the SNOW2.0 generator as defined in ISO/IEC 18033-4:2011 [6]. The Strumok algorithm uses the 256-bit initialization vector *IV* and the 256 or 512-bit secret key *K* and provides high and

ultra-high resistance, taking into the possible using account of quantum cryptographic analysis. The crypto algorithm is oriented on 64-bit computing systems, so the word size is set to 64 bits.

The main generator components are the linear feedback shift register (*LFSR*) and the finite-state machine (*FSM*), in which performed a non-linear transformation. The input data are used for initializing the variable state $S_i = (s^{(i)}, r^{(i)})$, $i \geq 0$, which consists of eighteen 64-bit blocks (words) that has two components: 16 variables $s^{(i)}$ – words of the *LFSR*: $s^{(i)} = (s_{15}^{(i)}, s_{14}^{(i)}, ..., s_0^{(i)})$; two words of the FSM $r^{(i)}$: $r^{(i)} = (r_2^{(i)}, r_1^{(i)})$. On the output, it gets a keystream, which formed with 64-bit words Z_i.

A schematic operation representation of keystream Strumok generator is in an arbitrary moment of the time *i*, it is represented in Fig. 1.

The feedback taps in the *LFSR* are constructed over the finite field $GF(2^{64})$ by a primitive polynomial: $f(x) = x^{16} + x^{13} + \alpha^{-1}x^{11} + \alpha$, where α is the root over the finite field $GF(2^8)$ of the primary polynomial:

$$g(z) = z^8 + \beta^{170}z^7 + \beta^{166}z^6 + \beta^2 z^5 + \beta^{224}z^4 + \beta^{70}z^3 + \beta^2.$$

In turn, the finite field $GF(2^8)$ is constructed over a $GF(2)$ field by a primitive polynomial:

$$p(y) = y^8 + y^4 + y^3 + y^2 + 1,$$

and the polynomial coefficients $g(z)$ are submitted through degree of a primitive element β of the finite field $GF(2^8)$ that is β – the root of a polynomial $p(y)$ Thus, we have an extension:

$$GF(2) \subset GF(2^8) \subset GF(2^{64}) \subset GF(2^{1024}),$$

where: the finite field $GF(2^{1024})$ is given by output of feedback the *LFSR* as a quotient ring $GF(2^{64})[x]/(f(x))$; the finite field $GF(2^{64})$ is given as a quotient ring $GF(2^8)[z]/(g(z))$; the finite field $GF(2^8)$ is given as a quotient ring $GF(2)[y]/(p(y))$. Therefore, output sequence period of the *LFSR* is maximal and equals of $2^{1024} - 1$.

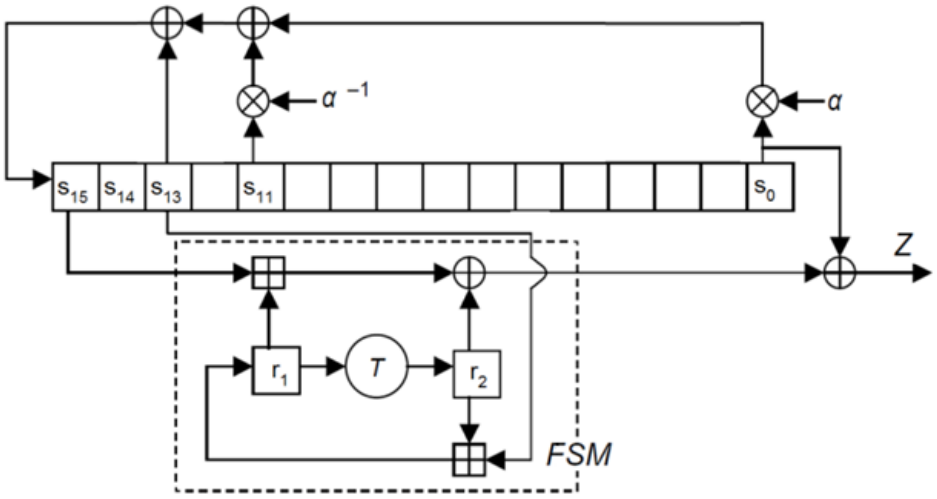


Fig.1 - Conceptual scheme of the *"Strumok"* keystream generator in generating gamma cipher mode (a keystream)

Structurally in the Strumok algorithm, it can distinguish three main functions:

- The initializing function *Init* that takes the key *K* (256 or 512 bits) and initialization vector *IV* (256 bits) as the input data, and produces the initial value of the variable state $S_0 = (s^{(0)}, r^{(0)})$;
- The next-state function *Next*, which takes the variable state S_i into the input and produces the next value of the variable state $S_{i+1} = (s^{(i+1)}, r^{(i+1)})$. The *Next* function can be in two modes,

which depending on how the iteration performed – as part of the implementation or as normal mode of generating output data.

- The keystream function *Strm* that takes the variable state S_i into the input and produces at the output 64-bit a keystream Z_i.

3 Initialization function Init

The initializing function of the internal state is described as follows.

Input: 256 or 512-bit a key *K*, 256-bit an initialization vector *IV*.

Output: the initial value of the variable state $S_0 = (s^{(0)}, r^{(0)})$.

The key for the stream cipher version Strumok-256 can represented as four 64-bit words $K = (K_3, K_2, K_1, K_0)$ and for the 512-bit a key – in the form $K = (K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0)$, where K_3 and K_7 s respectively 256 and 512 bits, the most significant words, and K_0 – the least significant.

The initialization vector can represented in the form of four 64-bit words $IV = (IV_3, IV_2, IV_1, IV_0)$, where IV_3 and IV_0 is respectively the most and the least significant words.

1. It is entered the key value in 16 words of the *LFSR*.

For a 256-bit version, the following operations are performed:

$$s_{15}^{(-33)} = \neg K_0,\ s_{14}^{(-33)} = K_1,\ s_{13}^{(-33)} = \neg K_2,\ s_{12}^{(-33)} = K_3,\ s_{11}^{(-33)} = K_0,\ s_{10}^{(-33)} = \neg K_1,\ s_9^{(-33)} = K_2,\ s_8^{(-33)} = K_3,$$
$$s_7^{(-33)} = \neg K_0,\ s_6^{(-33)} = \neg K_1,\ s_5^{(-33)} = K_2 \oplus IV_3,\ s_4^{(-33)} = K_3,\ s_3^{(-33)} = K_0 \oplus IV_2,\ s_2^{(-33)} = K_1 \oplus IV_1,$$
$$s_1^{(-33)} = K_2,\ s_0^{(-33)} = K_3 \oplus IV_0.$$

For a 512-bit version:

$$s_{15}^{(-33)} = K_0,\ s_{14}^{(-33)} = \neg K_1,\ s_{13}^{(-33)} = K_2,\ s_{12}^{(-33)} = K_3,\ s_{11}^{(-33)} = \neg K_7,\ s_{10}^{(-33)} = K_5,\ s_9^{(-33)} = \neg K_6,$$
$$s_8^{(-33)} = K_4 \oplus IV_3,\ s_7^{(-33)} = \neg K_0,\ s_6^{(-33)} = K_1,\ s_5^{(-33)} = K_2 \oplus IV_2,\ s_4^{(-33)} = K_3,\ s_3^{(-33)} = K_4 \oplus IV_1,$$
$$s_2^{(-33)} = K_5,\ s_1^{(-33)} = K_6,\ s_0^{(-33)} = K_7 \oplus IV_0$$

2. It performs 32 tacts of triggering without generating a key stream, i.e. two full cycles. Formally, it is presented as follows: $S_{-1} = Next^{32}(S_{-33}, INIT)$ which means 32 iterations to perform the *Next* function in the initialization mode *INIT*, $S_{-33} = (s^{(-33)}, r^{(-33)})$ the values of the variable state are calculated in previous step.

3. The initial value of the variable state S_0 calculated, according to the rule: $S_0 = Next(S_{-1})$ i.e. by executing the *Next* function in normal mode.

4. Get the output value of the variable state S_0.

4 Next-state function Next

The function of next-state, is described as follows.

Input: the variable state $S_i = (s^{(i)}, r^{(i)})$ selected mode (normal or initialization mode).

Output: the next value of the variable state $S_{i+1} = (s^{(i+1)}, r^{(i+1)})$.

1. A nonlinear substitution performed to update the value of the word $r_2^{(i+1)}$ the FSM. For this value, the *T* function is calculated: $r_2^{(i+1)} = T(r_1^{(i)})$.

2. The value of the word $r_1^{(i+1)}$ FSM is updated. For this value calculated, as following: $r_1^{(i+1)} = r_2^{(i+1)} +_{64} s_{13}^{(i)}$, where $+_{64}$ denotes the operation of adding integers by modulus 2^{64} (in the scheme of the cipher, see Fig. 1 this operation is marked as ⊞).

3. The 15 words value *LFSR* is updated $s_j^{(i+1)} = s_{j+1}^{(i)}$ where $j = 0, 1, \ldots, 14$.

4. The value of the 16-th word the *LFSR* is updating. If the normal mode the Next function, the value of this word computing by the rule:

$$s_{15}^{(i+1)} = (s_0^{(i)} \otimes \alpha) \oplus (s_{11}^{(i)} \otimes \alpha^{-1}) \oplus s_{13}^{(i)}.$$

If initialization mode *INIT* of the *Next* function is set, the value is computing by the rule:

$$s_{15}^{(i+1)} = FSM(s_{15}^{(i)}, r_1^{(i)}, r_2^{(i)}) \oplus (s_0^{(i)} \otimes \alpha) \oplus (s_{11}^{(i)} \otimes \alpha^{-1}) \oplus s_{13}^{(i)}.$$

The multiplication operations $\otimes$ on α and on α^{-1}, as well as the essence of the *FSM* function explained further.

5. The variable state $S_i = (s^{(i)}, r^{(i)})$ value is calculated and outputs.

The conceptual scheme of keystreams "Strumok" generator, when performing the *Next* function in the initialization mode is shown in Fig. 2.

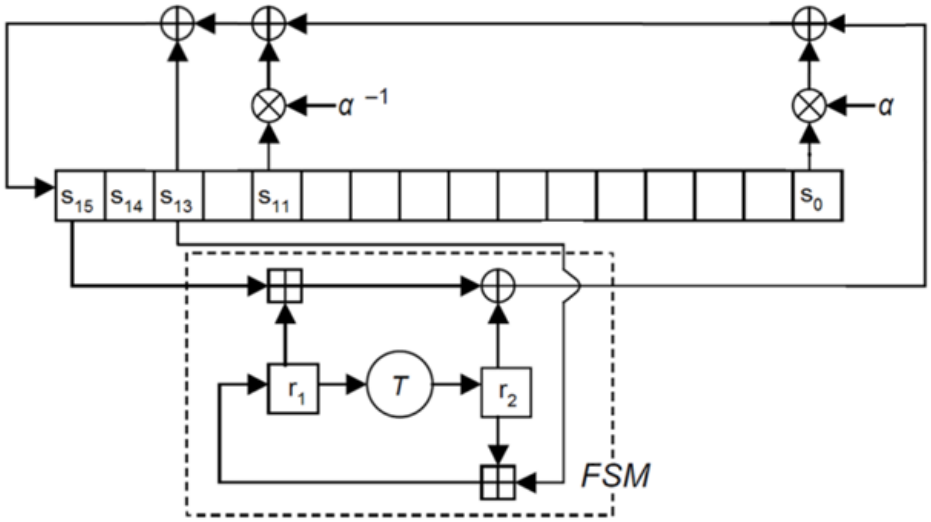


Fig.2 - The conceptual scheme of keystreams *"Strumok"* generator in initialization mode of the *Next* function

5 Keystream function Strm

Input: the variable state $S_i = (s^{(i)}, r^{(i)})$.

Output: 64-bit a key stream Z_i

1. The value is calculated

$$Z_i = FSM(s_{15}^{(i)}, r_1^{(i)}, r_2^{(i)}) \oplus s_0^{(i)}.$$

2. Get the output value Z_i.

6 Function finite-state machine FSM

The finite-state machine function is marked as *FSM (x,y,z)* and described as following.

Input: three 64-bit words *x*, *y* and *z*.

Output: 64-bit word *q*.

1. The value is calculated $q = (x +_{64} y) \oplus z$.

2. Get the output value *q*.

7 Function nonlinear substitution T

The nonlienar substitution of the *T* function implements finite field $GF(2^{64})$ rearrangment elements using the components of the block symmetric cryptographic transformation national standard DSTU 7624:2014 [19].

Input: 64-bit word *w*.
Output: 64-bit word $T = T(w)$.

1. The input word w is divided into 8-bit sub-blocks w_j: $w = (w_7, w_6, ..., w_0)$.

2. For each sub-blocks w_j is performed the DSTU 7624:2014 algorithm substitution using four table transformation $\pi_0, \pi_1, \pi_2, \pi_3$. As a result an output vector formed $r = (r_7, r_6, ..., r_0)$, $r_j = \pi_{j \bmod 4}[w_j]$, where $j = 0, 1, \ldots, 7$.

3. The vector calculated $q = (q_7, q_6, ..., q_0)$ using the rule (as in DSTU 7624:2014): $q_i = (\upsilon >>> i) r^T$, where υ given in hexadecimal form and $\upsilon = (01, 01, 05, 01, 08, 06, 07, 04)$, $>>> i$ - a cyclic shift operation on i bits to the right, $i = 0, 1, ..., 7$, $r^T = (r_0, r_1, ..., r_7)^T$ and the elements of the vectors r and q interpreted as elements of the final field $GF(2^8)$, which is given as a quotient ring $GF(2)[y]/(p(y))$.

4. Get the output value q, which interpreted as a 64-bit word.

The vector quick calculation $(q_0, q_1, ..., q_7) = Q$ is implemented by the rule:

$$Q^T = T_0[w_0] \oplus T_1[w_1] \oplus T_2[w_2] \oplus T_3[w_3] \oplus T_4[w_4] \oplus T_5[w_5] \oplus T_6[w_6] \oplus T_7[w_7],$$

where:

$$T_0[a] = \begin{pmatrix} 01 \\ 04 \\ 07 \\ 06 \\ 08 \\ 01 \\ 05 \\ 01 \end{pmatrix} \cdot \pi_0[a],\ T_1[a] = \begin{pmatrix} 01 \\ 01 \\ 04 \\ 07 \\ 06 \\ 08 \\ 01 \\ 05 \end{pmatrix} \cdot \pi_1[a],$$

$$T_2[a] = \begin{pmatrix} 05 \\ 01 \\ 01 \\ 04 \\ 07 \\ 06 \\ 08 \\ 01 \end{pmatrix} \cdot \pi_2[a],\ T_3[a] = \begin{pmatrix} 01 \\ 05 \\ 01 \\ 01 \\ 04 \\ 07 \\ 06 \\ 08 \end{pmatrix} \cdot \pi_3[a],$$

$$T_4[a] = \begin{pmatrix} 08 \\ 01 \\ 05 \\ 01 \\ 01 \\ 04 \\ 07 \\ 06 \end{pmatrix} \cdot \pi_0[a],\ T_5[a] = \begin{pmatrix} 06 \\ 08 \\ 01 \\ 05 \\ 01 \\ 01 \\ 04 \\ 07 \end{pmatrix} \cdot \pi_1[a],$$

$$T_6[a]=\begin{pmatrix}07\\06\\08\\01\\05\\01\\01\\04\end{pmatrix}\cdot\pi_2[a],\ T_7[a]=\begin{pmatrix}04\\07\\06\\08\\01\\05\\01\\01\end{pmatrix}\cdot\pi_3[a].$$

The constant tables $T_i[a]$, $i=0,1,...,7$ using enables significantly reducing the operations number, in particular the nonlinear substitution function, is calculated seven XOR operations over 64-bit words.

8 Multiplications of α in $GF(2^{64})$

Multiplication on α in the finite field $GF(2^{64})$ arithmetic is implemented by a table of pre-calculus Mul_α, which contains 256 rows of 64 bits in each.

1. The value is calculated

$$w'=(w<<8)\oplus Mul_\alpha[w>>56] \quad (1)$$

where:

- $w<<8$ is the result of a shift to the left (towards the higher significant bits) 64-bit word w on 8 bits with the filling of the less significant bits with zero values.
- $w>>56$ is the result of a shifting to the right (*towards the less significant bits*) 64-bit word w on 56 bits with the filling of the higher significant bits with zero values. Eight less bits of the vector $w>>56$ is interpreted as the finite field $GF(2^8)$ element for indexing the table of pre-calculus Mul_α;
- $Mul_\alpha[c]$ - 64-bit value the table of pre-calculus in the row with the index $c\in GF(2^8)$, where $Mul_\alpha[c]\in GF(2^{64})$.

2. Get the output value w'.

9 Multiplications of α^{-1} in $GF(2^{64})$

Multiplication on α^{-1} in arithmetic of the finite field $GF(2^{64})$ is implemented by a table of pre-calculus $Mul_{\alpha^{-1}}$, which contains 256 rows of 64 bits in each.

1. The value is calculated

$$w'=(w>>8)\oplus Mul_{\alpha^{-1}}[w\&\gamma] \quad (2)$$

where:

- $w>>8$ is the result of a shifting to the right (towards the less significant bits) 64-bit word w on 8 bits with the filling of the higher significant bits with zero values.
- $w\&\gamma$ is a bitwise conjunction result of words w and γ, which is in the hexadecimal form $\gamma=$ 00000000000000FF. The eight less significant bits of the vector $w\&\gamma$ is interpreted as an element of the finite field $GF(2^8)$ for indexing the table of pre-calculus $Mul_{\alpha^{-1}}$.

3. Get the output value w'.

10 Value tables-constant Mul_α and $Mul_{\alpha^{-1}}$

For the fast encryption there are used tables of pre-calculus Mul_α and $Mul_{\alpha^{-1}}$. This allows significantly reduce the number of operation to handle the input data.

The polynomial that defines feedback of the LFSR has the expression $g(z)$. Thus, each word of the LFSR stores a 64-bit sequence w, which is represented as eight sub-blocks w_j of 8 bits in each: $w=(w_7, w_6, ..., w_0)$, which is interpreted as coefficients of a polynomial $w(z) \in GF(2^8)[z]/(g(z))$.

If $\alpha = z$ is a root with a primitive polynomial of the $GF(2^8)$:

$$g(z) = z^8 + g_7 z^7 + ... + g_0$$

then:

$$w(z) \cdot \alpha = w_{<<8}(z) + w_{>>56}(z) \cdot g'(z),$$

where $w_{<<8}(z)$, $w_{>>56}(z)$ and $g'(z)$ are polynomials, and have the form:

$$w_{<<8}(z) = w_6 z^7 + w_5 z^6 + ... + w_0 z, \; w_{>>56}(z) = w_7,$$
$$g'(z) = g_7 z^7 + g_6 z^6 + ... + g_0.$$

The binary representation of the polynomials $w_{<<8}(z)$ and $w_{>>56}(z)$ forms coefficients is considered in (1) binary sequences $w<<8$ and $w>>56$. So computing $w(z) \cdot \alpha$ in the finite field $GF(2^{64})$ arithmetic corresponds to the formula (1), where 256 values of the table $Mul_{\alpha}[w_7]$ is calculated as 64-bit sequences with the binary representation of coefficients $(w_7 g_7, w_7 g_6, ..., w_7 g_0)$ a polynomial

$$w_{>>56}(z) \cdot g'(z) = w_7 (g_7 z^7 + g_6 z^6 + ... + g_0)$$

for each with 256 possible values $w_7 \in GF(2^8)$.

If $\alpha = z$ then:

$$\alpha^8 = g_7 \alpha^7 + ... + g_1 \alpha + g_0 \alpha^0,$$

so

$$g_0^{-1} \alpha^7 + g_0^{-1} g_7 \alpha^6 + ... + g_0^{-1} g_1 \alpha^0 = \alpha^{-1} = z^{-1},$$

then

$$w(z) \alpha^{-1} = w_{>>8}(z) + w_0(z) \cdot g''(z),$$

where $w_{>>8}(z)$, $w_0(z)$ and $g''(z)$ are polynomials, and have the form:

$$w_{>>8}(z) = w_7 z^6 + w_6 z^5 + ... + w_1,$$
$$w_0(z) = w_0,$$
$$g''(z) = g_0^{-1} z^7 + g_0^{-1} g_7 z^6 + ... + g_0^{-1} g_1.$$

The binary representation of the polynomials coefficients $w_{>>8}(z)$ and $w_0(z)$ forms is considered in (2) binary sequences $w>>8$ and γ. So computing $w(z) \cdot \alpha^{-1}$ in arithmetic of the finite field $GF(2^{64})$ corresponds to formula (2), where 256 values of the table $Mul_{\alpha^{-1}}[w_0]$ is calculated as 64-bit sequences with the binary representation of coefficients $(w_0 g_0^{-1}, w_0 g_0^{-1} g_7, ..., w_0 g_0^{-1} g_1)$ a polynomial

$$w_0(z) \cdot g''(z) = w_0 (g_0^{-1} z^7 + g_0^{-1} g_7 z^6 + ... + g_0^{-1} g_1)$$

for each with 256 possible values $w_0 \in GF(2^8)$.

11 Software performance of "*Strumok*"

For the generation key stream rate research, we realized experiment as the well-known symmetric cryptographic transformation on equal terms. List of algorithm, source of specification and brief information are given in the table 1. The testing results based on the criterion for long streams encryption [20] are shown in the table 2. As we can see from the data in the table, the keystream "Strumok" generator enables pseudo-random sequences forming with speeds exceeding of 10 Gbit

per sec. By this measure, it is ahead of almost all the most common ciphers including the algorithm SNOW 2.0.

Table 1 – List of algorithm

Names cipher	Source of specification	State size, bit	Key size, bit	Size IV, bit
AES	FIPS-197, CRYPTREC, ISO/IEC 18033-4	128	128, 256	256
Kalyna	DSTU 7624:2014	128, 256, 512	128, 256, 512	128, 256, 512
HC	eSTREAM	128, 256	128, 256	128, 256
MICKEY	eSTREAM	160	128	128
RABBIT	ISO/IEC 18033-4, eSTREAM	513	128	64
SALSA-20	eSTREAM	512	128	64
SNOW2.0	ISO/IEC 18033-4	512	128, 256	128, 56
SOSEMANUK	eSTREAM	512	128	128
TRIVIUM	eSTREAM, ISO/IEC 29192-3	288	80	80
Enocoro	ISO/IEC 29192-3	272	80, 128	64
CRYPTMT3	eSTREAM	128	128	64
DECIMv2	ISO/IEC 18033-4, eSTREAM	288	128	128
RC4	Список рассылки Cypherpunks	256	256	–
KCIPHER-2	ISO/IEC 18033-4, CRYPTREC	640	128	128
GRAIN	eSTREAM	128	128	96
MUGI	ISO/IEC 18033-4	128	128	128
Strumok-256	This article	1024	256	256
Strumok-512			512	

Table 2 – Performance evaluation of ciphers

CIPHERS	Intel Core i7-6820HQ 2.7 Gh	Intel Core i7-5500u 2.4 Gh	Intel Pentium P6200 2.13 Gh
AES-128	2,48	1,75	1,12
AES-256	1,66	1,18	0,80
Kalyna-128	2,56	1,79	0,83
Kalyna-256	1,71	1,21	0,57
Kalyna-512	1,42	0,99	0,46
HC-128	11,46	7,69	4,25
HC-256	5,13	3,88	2,03

a continuation Table 2

CIPHERS	Intel Core i7-6820HQ 2.7 Gh	Intel Core i7-5500u 2.4 Gh	Intel Pentium P6200 2.13 Gh
MICKEY-128	0,07	0,05	0,03
RABBIT	3,65	2,77	1,64
SALSA-20	3,02	2,06	1,41
SNOW2.0-128	8,76	5,43	3,67
SNOW2.0-256	8,72	5,54	3,59
SOSEMANUK	4,07	2,56	1,82
TRIVIUM	3,89	2,78	1,89
CryptMT3	5,92	4,63	4,04
DECIM-128	0,01	0,01	0,01
RC4	3,58	3,21	1,67
KCIPHER-2	0,40	0,40	0,31
GRAIN	0,01	0,01	0,00
MUGI	3,62	2,98	2,58
Strumok-256	13,31	10,04	5,10
Strumok-512	13,70	9,74	5,08

Conclusions

Strumok in its conceptual scheme similar to the *SNOW 2*.0. But *SNOW 2*.0 focused on the use of 32-bit computing systems, while "*Strumok"* is intended for use in more powerful 64-bit computing systems. With this in "*Strumok"* increased rate of formation of the pseudo-random sequence, as used by 64-bit words to store encryption keystream. The conducted comparative tests have shown that the "*Strumok"* on 32-bit computing systems also shows good performance results. Using a pre-computation increases the speed of the algorithm, since there is no need to make complicated calculations during the generation of the keystream.

In the "*Strumok"* algorithm, compared with SNOW 2.0, it is increased the length of the secret key and the initialization vector. This allows us reliably apply a stream cipher even with taking into account quantum methods of cryptographic analysis. Thus, in aggregate of properties, the "*Strumok"* can be considered as a candidate for the symmetric encryption national standard of Ukraine.

References

[1] N. Ferguson and B. Schneier. Practical Cryptography. John Wiley & Sons, 2003, 432 p.

[2] A.J. Menezes, P.C. van Oorschot, S.A. Vanstone. Handbook of Applied Cryptography. CRC Press, 1997, 794 p.

[3] N. Koblitz and A.J. Menezes. "A Riddle Wrapped in an Enigma". Internet: https://eprint.iacr.org/2015/1018.pdf, Oct. 20, 2015 [Aug. 21, 2016]

[4] D. Bernstein, J. Buchmann and E.Dahmen. Post-Quantum Cryptography. Springer-Verlag, Berlin-Heidleberg, 2009, 245 p.

[5] D. Moody. "Post-Quntum Cryptography: NIST's Plan for the Future". The Seventh International Conference on Post-Quntum Cryptography, Japan, 2016. [On-line]. Internet: https://pqcrypto2016.jp/data/pqc2016_nist_announcement.pdf

[6] ISO/IEC 18033-4:2011. "Information technology – Security techniques – Encryption algorithms – Part 4: Stream ciphers". On-line]. Internet: http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=54532 [Dec., 2012]

[7] O. Kuznetsov, M. Lutsenko and D. Ivanenko, "Strumok stream cipher: Specification and basic properties". *2*016 Third International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T), Kharkiv, 2016, pp. 59-62.

[8] O. Kuznetsov, Y. Gorbenko and I. Kolovanova, "Combinatorial properties of block symmetric ciphers key schedule". 2016 Third International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T), Kharkiv, 2016, pp. 55-58.

[9] I. Gorbenko, A. Kuznetsov, M. Lutsenko and D. Ivanenko, "The research of modern stream ciphers". 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, 2017, pp. 207-210.
[10] A. Kuznetsov, I. Svatovskij, N. Kiyan and A. Pushkar'ov, "Code-based public-key cryptosystems for the post-quantum period". 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, 2017, pp. 125-130.
[11] A. Kuznetsov, I. Kolovanova and T. Kuznetsova, "Periodic characteristics of output feedback encryption mode". 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, 2017, pp. 193-198.
[12] A. Kuznetsov, Y. Gorbenko, A. Andrushkevych and I. Belozersev, "Analysis of block symmetric algorithms from international standard of lightweight cryptography ISO/IEC 29192-2". 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, 2017, pp. 203-206.
[13] Y. Izbenko, V. Kovtun and A. Kuznetsov, "The Design of Boolean Functions by Modified Hill Climbing Method". 2009 6th International Conference on Information Technology: New Generations, Las Vegas, NV, 2009, pp. 356-361.
[14] A. Kuznetsov, R. Serhiienko and D. Prokopovych-Tkachenko, "Construction of cascade codes in the frequency domain". 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, 2017, pp. 131-136.
[15] A. Andrushkevych, T. Kuznetsova, I. Bilozertsev and S. Bohucharskyi, "The block symmetric ciphers in the post-quantum period". 2016 Third International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T), Kharkiv, 2016, pp. 43-46.
[16] I.D. Gorbenko, V.I. Dolgov, V.I. Rublinetskii, K.V. Korovkin. “Methods of Information Protection in Communications Systems and Methods of Their Cryptoanalysis”. Telecommunications and Radio Engineering, Vol. 52, Issue 4, (1998), pp. 89-96.
[17] I. Gorbenko, V. Ponomar. “Examining a possibility to use and the benefits of post-quantum algorithms dependent on the conditions of their application”. EasternEuropean Journal of Enterprise Technologies, Vol. 2, No. 9 (86) (2017), pp. 21-32.
[18] Yu.V. Stasev, A.A. Kuznetsov. “Asymmetric code-theoretical schemes constructed with the use of algebraic geometric codes”. Kibernetika i Sistemnyi Analiz, No. 3, pp. 47-57, May-June 2005.
[19] DSTU 7624:2014. “Informacijni tehnologii'. Kryptografichnyj zahyst informacii'. Algorytm symetrychnogo blokovogo peretvorennja”. (in Ukrainian). [On-line]. Internet: http://shop.uas.org.ua/ua/informacijni-tehnologii-kriptografichnij-zahist-informacii-algoritm-simetrichnogo-blokovogo-peretvorennja.html
[20] “eSTREAM Optimized Code HOWTO”. [On-line]. Internet: http://www.ecrypt.eu.org/stream/perf/ [Nov. 1, 2005].

Рецензент: Микола Карпінський, д.т.н., проф., університет Бельсько-Бяла, Польща.
E-mail: mkarpinski@ath.bielsko.pl

Надійшло: Лютий 2018.

Автори:
Іван Горбенко, д.т.н., проф., Харківський національний університет імені В.Н. Каразіна, м. Харків, Україна.
E-mail: gorbenkoi@iit.kharkov.ua
Олександр Кузнецов, д.т.н., проф., академік Академії наук прикладної радіоелектроніки, ХНУ імені В.Н. Каразіна, м. Харків, Україна. E-mail: kuznetsov@karazin.ua
Юрій Горбенко, к.т.н., акціонерне товариство “Інститут інформаційних технологій” (АТ "ІІТ"), м. Харків, Україна.
E-mail: gorbenkou@iit.kharkov.ua
Антон Олексійчук, д.т.н., національний технічний університет України "КПІ", (НТУУ "КПІ"), м. Київ , Україна.
E-mail: alex-dtn@ukr.net
Владислав Тімченко, студент факультету комп’ютерних наук, Харківський національний університет імені В.Н. Каразіна, м. Харків, Україна.
E-mail: tvlad.tyma@gmail.com

Потоковий шифр "Струмок".

Анотація. У роботі представлені основні результати розробки нового генератора ключів «Струмок», який пропонується в якості кандидата на національний стандарт симетричного шифрування України. «Струмок» побудований на схемі SNOW 2.0-like генератора підсумовування. Збільшена довжина секретного ключа і вектор ініціалізації дозволяють надійно використовувати шифр потоку навіть з урахуванням методів квантового криптографічного аналізу. На відміну від SNOW 2.0 «Strumok» призначений для використання в більш потужних 64-розрядних обчислювальних системах. Проведені порівняльні тести показали, що «Strumok» на 32-розрядних обчислювальних системах також показує хороші результати. Представлені основні перетворення і окремі результати дослідження продуктивності шифрування, розглянуто генератор, якій забезпечує формування потоку ключів зі швидкістю, що перевищує 10 Гбіт / сек.

Ключові слова: шифрування; потоковий шифр; синхронний генератор ключів; псевдовипадкова послідовність.

Рецензент: Николай Карпинский д.т.н., проф., университет Бельсько-Бяла, Польша.
E-mail: mkarpinski@ath.bielsko.pl

Поступила: Февраль 2018.

Авторы:

Иван Горбенко, д.т.н., проф., Харьковский национальный университет имени В.Н. Каразина, г. Харьков, Украина.
E-mail: gorbenkoi@iit.kharkov.ua

Александр Кузнецов, д.т.н., проф., академик Академии наук прикладной радиоэлектроники, ХНУ имени В.Н. Каразина, г. Харьков, Украина. E-mail: kuznetsov@karazin.ua

Юрий Горбенко, к.т.н., АТ “Институт информационных технологий” (АТ "ИИТ"), г. Харьков, Украина.
E-mail: gorbenkou@iit.kharkov.ua

Антон Алексейчук, д.т.н., национальный технический университет Украины "КПИ", (НТУУ "КПИ"), г. Киев, Украина.
E-mail: alex-dtn@ukr.net

Владислав Тимченко, студент факультета компьютерных наук, Харьковский национальный университет имени В.Н. Каразина, г. Харьков, Украина.
E-mail: tvlad.tyma@gmail.com

Потоковый шифр "Струмок".

Аннотация. В работе представлены основные результаты разработки нового генератора ключей «Струмок», который предлагается в качестве кандидата на национальный стандарт симметричного шифрования Украины. «Струмок» построен на схеме SNOW 2.0-like генератора суммирования. Увеличенная длина секретного ключа и вектор инициализации позволяют надежно использовать шифр потока даже с учетом методов квантового криптографического анализа. В отличие от SNOW 2.0, «Струмок» предназначен для использования в более мощных 64-разрядных вычислительных системах. Проведенные сравнительные тесты показали, что «Струмок» на 32-разрядных вычислительных системах также показывает хорошие результаты. Представлены основные преобразования и отдельные результаты исследования производительности шифрования, рассмотрен генератор, обеспечивающий формирование потока ключей со скоростью, превышающей 10 Гбит / сек.

Ключевые слова: шифрование; потоковый шифр; синхронный генератор ключей; псевдослучайная последовательность.

UDC 681.142.01

A CONCEPTION FOR COMPARISON OF INTEGER DATA REPRESENTED IN A RESIDUE NUMBER SYSTEM

Viktor Krasnobayev[1], Sergey Koshman[1], Alina Yanko[2], Sergey Moroz[3]

[1] V. N. Karazin Kharkiv National University, 4 Svobody Sq., Kharkiv, 61022, Ukraine
v.a.krasnobaev@gmail.com, s_koshman@ukr.net
[2] Poltava National Technical Yuri Kondratyuk University, 24 Pershotravnevyi Av., Poltava, 36011, Ukraine
al9_yanko@ukr.net
[3] Kharkiv Petro Vasylenko National Technical University of Agriculture, 19 Rizdviana St., Kharkiv, 61052, Ukraine
frost9i@ukr.net

Reviewer: Sergii Kavun, Doctor of Sciences (Economics), Ph.D. (Engineering), Full Prof., Kharkiv University of Technology "STEP", Kharkiv, 61010 , Ukraine
kavserg@gmail.com

Received on March 2018

Abstract: *The methods for comparison integer data that are represented in the residue number system (RNS) are described. The method of arithmetic comparison of integer data is developed, which improves the accuracy of processing of information presented in the RNS. The developed mathematical model and the method of precise arithmetic comparison of data in RNS, which are based on obtaining and using the positional feature of the non-position code, provide maximum reliability of the result of comparing numbers with a minimum amount of equipment of the comparator. The use of the developed method makes it possible to increase the efficiency of the operation of specialized computing devices in the RNS. Based on the developed method, a device was synthesized for the implementation of the comparison process in the RNS to which the patent of Ukraine was obtained, which confirms the novelty of the world and the practical significance of the results of this article.*

Keywords: *data processing system; residue number system; arithmetic integer comparison of data; accuracy processing of data; nulevization of number.*

1 Introduction

As is well known, the prime advantage of a position-independent residue number system (RNS) is the possibility to organize the process of fast processing of integers. The use of RNSs makes it possible to create methods and digital hardware of computer systems that improve user efficiency in solving definite classes of problems in which the operations of integer arithmetic addition, subtraction, and multiplication are applied. This is reached owing to the use of RNS properties such as independence, equality, and small length of residues whose totality $\{a_i\}$ represents a number $A_{RNS} = (a_1 \| a_2 \| ... \| a_{i-1} \| a_i \| a_{i+1} \| ... \| a_n)$ in terms of n bases (moduli) of a given position-independent number system [1,2].

The need for solving a wide class of problems containing logical operations (*for example, the operation of comparison* $A_{RNS} = (a_1 \| a_2 \| ... \| a_{i-1} \| a_i \| a_{i+1} \| ... \| a_n)$ *and* $B_{RNS} = (b_1 \| b_2 \| ... \| b_{i-1} \| b_i \| b_{i+1} \| ... \| b_n)$, which often occurs in control problems) along with arithmetic integer operations by a computer system that processes integer data (CSPID) and perform operations on RNS numbers reduces the overall efficiency of using a position-independent number system. This is stipulated by a considerable (*in comparison with the execution of the above-mentioned arithmetic modular operations*) time of execution of the data comparison operation in RNS. Therefore, the investigation and improvement of the existing methods and algorithms for hardware implementation of the operation of arithmetic comparison of data in RNS and also the development of new ones is an important and topical scientific and applied problem of creation of CSPIDs.

2 Methods for comparing numbers in a residue number system

As is well known, there are three groups of methods for comparing numbers in RNS [3,4]. The methods of direct comparison belong to the first group. They are based on the transformation of numbers A_{RNS} and B_{RNS} from an RNS code into the positional binary number system (PNS) $A_{PNS} = \overline{\alpha_\rho, \alpha_{\rho-1}....\alpha_1}$ and $B_{PNS} = \overline{\beta_\rho, \beta_{\rho-1},.... \beta_1}$ (ρ *is the digit capacity of the numbers* A_{PNS} *and* B_{PNS}) and their further comparison using binary positional adders. The methods based on the principle of nulevization belong to the second group. The procedure of the process of nulevization consists of the passage from the initial number $A_{RNS} = (a_1 \| a_2 \| ... \| a_{i-1} \| a_i \| a_{i+1} \| ... \| a_n)$ represented in an RNS to a number of the form $A^{(u)}{}_{RNS} = [0 \| 0 \| ... \| 0 \| \gamma_n^{(A_{RNS})}]$. With the help of the value of $\gamma_n^{(A_{RNS})}$, the numerical interval $[j_{A_{RNS}} m_n, (j_{A_{RNS}} + 1)m_n)$ is determined that contains the number A_{RNS}. The nulevization of the number $B_{RNS} = (b_1 \| b_2 \| ... \| b_{i-1} \| b_i \| \| b_{i+1} \| ... \| b_n)$ is similarly performed, whence we obtain the values of $\gamma_n^{(B_{RNS})}$. In this case, the value of $\gamma_n^{(B_{RNS})}$ determines the numerical interval $[j_{B_{RNS}} m_n, (j_{B_{RNS}} + 1)m_n)$ that contains the numbers B_{RNS}. The result of the operation of arithmetic comparison of the numbers A_{RNS} and B_{RNS} in RNS is determined by comparing the obtained values of $\gamma_n^{(A_{RNS})}$ and $\gamma_n^{(B_{RNS})}$ or the values of the corresponding quantities $j_{A_{RNS}}$ и $j_{B_{RNS}}$ ($j_{A_{RNS}}$, $j_{B_{RNS}} = \overline{0, \prod_{i=1}^{n-1} m_i}$).

The methods based on the determination of existing features or additional formation and use of special indicators of a number in RNS, i.e., the so-called position signs of a position-independent code (PIPIC) in RNS belong to the third group. Such signs (*for example, a rank r of a number*) contain information on the values of the numbers being compared and can be used for determining the value of a number in RNS. The use of PIPICs makes it possible to reduce the time of execution of the data comparison procedure in RNS in comparison with the first and second groups of methods.

The main drawback of the existing fast methods of arithmetic data comparison in RNS that are based on the use of PIPICs is the impossibility of ensuring the maximum accuracy in all cases of comparison of two numbers (A_{RNS} and B_{RNS}). This circumstance stipulates the obtainment of an uncertain result of comparison of numbers. The objective of this article is the development of a method for exact arithmetic comparison of two numbers $A_{RNS} = (a_1 \| a_2 \| ... \| a_{i-1} \| a_i \| a_{i+1} \| ... \| a_n)$ and $B_{RNS} = = (b_1 \| b_2 \| ... \| b_{i-1} \| b_i \| b_{i+1} \| ... \| b_n)$ in RNS on the basis of using PIPICs. The use of the proposed method of exact comparison of data will make it possible to reliably determine the result of the operation of arithmetic integer comparison of two numbers in RNS.

3 Method and algorithm for arithmetic comparison of numbers in RNS

We will briefly consider the essence of an existing method of arithmetic comparison of numbers $A_{RNS} = (a_1 \| a_2 \| ... \| a_{i-1} \| a_i \| a_{i+1} \| ... \| a_n)$ and $B_{RNS} = (b_1 \| b_2 \| ... \| b_{i-1} \| b_i \| b_{i+1} \| ... \| b_n)$ in RNS on the basis of using PIPICs. Let an RNS be specified by a collection $\{m_i\}$, $i = \overline{1,n}$ of pairwise primes. The greatest common divisor (GCD) of any pair of bases m_i and m_j ($i, j = \overline{1, n}$; $i \neq j$) is equal to one, i.e., GCD (m_i, m_j) = 1. For the sake of generality, we will consider that the RNS is ordered, i.e., $m_i < m_{i+1}$. The essence of the well-known method consists of the formation and use of PIPICs on the basis of constructing a special code (SC) for each of the numbers A_{RNS} and B_{RNS} being compared. In this case, for an arbitrary module m_i, the RNS for the numbers A_{RNS} and B_{RNS} being com-

pared is formed by a special code of the form $K_{N_{m_i}}^{(n_A)} = \{Z_{N_{m_i}-1}^{(A_{RNS})}\ Z_{N_{m_i}-2}^{(A_{RNS})} \dots Z_2^{(A_{RNS})} Z_1^{(A_{RNS})}\ Z_0^{(A_{RNS})}\}$ and $K_{N_{m_i}}^{(n_B)} = \left\{Z_{N_{m_i}-1}^{(B_{RNS})}\ Z_{N_{m_i}-2}^{(B_{RNS})} \dots Z_2^{(B_{RNS})}\ Z_1^{(B_{RNS})}\ Z_0^{(B_{RNS})}\right\}$.

The locations of zero bits $K_{N_{m_i}}^{(n_A)}$ and $K_{N_{m_i}}^{(n_B)}$ in the SC are determined by the PIPICs n_A and n_B of the numbers A_{RNS} and B_{RNS} respectively. The procedure of determining an SC is described in [5] in detail.

To understand the essence of the proposed comparison method, we will consider a geometrical interpretation of the process of comparison of two numbers. Figure 1 presents the scheme of partitioning a numerical interval [0, M) corresponding to the range of representation of the numbers $A_{RNS} = (a_1 \| a_2 \| \dots \| a_{i-1} \| a_i \| a_{i+1} \| \dots \| a_n)$ and $B_{RNS} = (b_1 \| b_2 \| \dots \| b_{i-1} \| b_i \| b_{i+1} \| \dots \| b_n)$ being compared, where $M = \prod_{i=1}^{n} m_i$. This numerical interval [0, M) is divided into N_{m_i} equal intervals $[jm_i, (j+1)m_i)$ of length m_i. The operation of transformation of the numbers A_{RNS} and B_{RNS} being compared by the use of the so-called nulevization constants $KH_{m_i}^{(A_{RNS})} = (a_1' \| a_2' \| \dots \| a_{i-1}' \| a_i \| a_{i+1}' \| \dots \| a_n')$ and $KH_{m_i}^{(B_{RNS})} = (b_1' \| b_2' \| \dots \| b_{i-1}' \| b_i \| b_{i+1}' \| \dots \| b_n')$ to the form $A_{m_i} = A_{RNS} - KH_{m_i}^{(A_{RNS})} = (a_1 \| a_2 \| \dots \| a_{i-1} \| a_i \| a_{i+1} \| \dots \| a_n) - (a_1' \| a_2' \| \dots \| a_{i-1}' \| a_i \| a_{i+1}' \| \dots \| a_n') = [a_1^{(1)} \| a_2^{(1)} \| \dots \| a_{i-1}^{(1)} \| 0 \| a_{i+1}^{(1)} \| \dots \| a_n^{(1)}]$ and $B_{m_i} = B_{RNS} - KH_{m_i}^{(B_{RNS})} = (b_1 \| b_2 \| \dots \| b_{i-1} \| b_i \| b_{i+1} \| \dots \| b_n) - (b_1' \| b_2' \| \dots \| b_{i-1}' \| b_i \| b_{i+1}' \| \dots \| b_n') = [b_1^{(1)} \| b_2^{(1)} \| \dots \| b_{i-1}^{(1)} \| 0 \| b_{i+1}^{(1)} \| \dots \| b_n^{(1)}]$ is equivalent the shift of these numbers A_{RNS} and B_{RNS} to the left edges of the corresponding numerical intervals $[j_1 m_i, (j_1+1)m_i)$ and $[j_2 m_i, (j_2+1)m_i)$ of their location, which corresponds to the reduction of them to numbers A_{m_i} and B_{m_i} that are multiple of the ith module m_i of the RNS. Then numbers $j_1 = n_A$ and $j_2 = n_B$ of these intervals are determined that, in this case, are position signs of a position-independent code in the RNS.

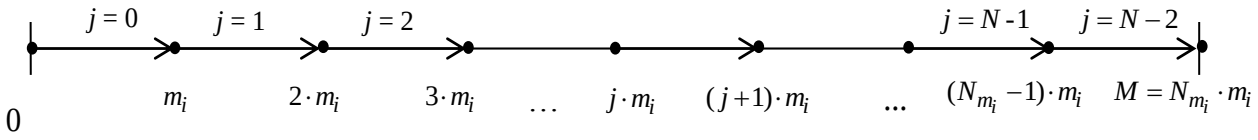


Fig. 1 – Scheme of partitioning a numerical interval [0, M) into equal intervals for an arbitrary base m_i of RNS

4 An indicator of an estimate for the accuracy of comparing numbers in RNS

As is well known, the most important characteristic of the process of comparison of numbers is the comparison accuracy W_{m_i}. In the general case, the accuracy W_{m_i} of comparison of two numbers $A_{RNS} = (a_1 \| a_2 \| \dots \| a_{i-1} \| a_i \| a_{i+1} \| \dots \| a_n)$ and $B_{RNS} = (b_1 \| b_2 \| \dots \| b_{i-1} \| b_i \| b_{i+1} \| \dots \| b_n)$ in RNS depends on locations of intervals $[j_1 m_i, (j_1+1)m_i)$ and $[j_2 m_i, (j_2+1)m_i)$ of these numbers on the numerical axis $0 \div M$, i.e., on the numbers j_1 and j_2 of ranges of these intervals. If $j_1 \neq j_2$, then the accuracy W_{m_i} of comparison of two numbers A_{RNS} and B_{RNS} depends only on the locations of the intervals $[j_1 m_i, (j_1+1)m_i)$ and $[j_2 m_i, (j_2+1)m_i)$ on the numerical axis $0 \div M$. The process of comparison of two A_{RNS} and B_{RNS} is as follows. If $A_{RNS} > B_{RNS}$, and if $j_1 < j_2$, then $A_{RNS} < B_{RNS}$. If $j_1 = j_2$, then $A_{RNS} = B_{RNS}$. In this case, the comparison accuracy W_{m_i} depends only on the range of

the interval $[j_1 m_i, (j_1+1)m_i)$ of location of the numbers A_{RNS} and B_{RNS}, i.e., on the value m_i of an RNS module. Proceeding from the aforesaid and also from the geometrical interpretation of the comparison process, it is obvious that, in the above case, the comparison accuracy W_{m_i} in RNS can be estimated using the relationship

$$W_{m_i} = 1/m_i . \quad (1)$$

Note that, for an arbitrary value m_i of an RNS module, the amount of equipment N_{m_i} of the device for comparing two numbers A_{RNS} and B_{RNS} that mainly depends on the amount of the equipment of two groups of adders entering in it and implementing the operations $A_{m_i} - K_A \cdot m_i = Z_{K_A}^{(A_{RNS})}$ and $B_{m_i} - K_B \cdot m_i = Z_{K_B}^{(B_{RNS})}$ is defined as follows:

$$N_{m_i} = \prod_{\substack{k=1; \\ k \neq i.}}^{n-1} m_k . \quad (2)$$

For large sizes of digit grids of CSPIDs, the value of $N_{m_i} = \prod_{\substack{k=1; \\ k \neq i.}}^{n-1} m_k$ can be rather sizeable.

Let us consider the process of comparison in the case when numbers $A_{RNS} = (a_1 \| a_2 \| ... \| a_{i-1} \| a_i \| a_{i+1} \| ... \| a_n)$ and $B_{RNS} = (b_1 \| b_2 \| ... \| b_{i-1} \| b_i \| b_{i+1} \| ... \| b_n)$ are in one interval $[jm_i, (j+1)m_i)$, i.e., $j_1 = j_2 = j$ ($A_{m_i} = B_{m_i} = j \cdot m_i$). Proceeding from the aforesaid, for this case of comparison, the numbers A_{RNS} and B_{RNS} will always be equal to each other, which is contrary to the facts in the majority of cases. To obtain a reliable result of comparison, the procedure of comparison of the numbers A_{RNS} and B_{RNS} should be additionally organized within the numerical interval $[jm_i, (j+1)m_i)$ itself to which they belong. To solve this problem (*with allowance made for the dependence on the magnitude of the module* m_i *being used*), we will consider variants of arithmetic comparison of numbers in RNS specified by an ordered collection $\{m_i\}$ $(i = \overline{1,n})$ of bases.

Variant 1. $m_i = m_n = \max$. In this case, the accuracy W_{m_n} of comparison in RNS is determined by the length m_n of the interval $[jm_n, (j+1)m_n)$, and it this length that will be minimal (see relationship (1)). In this case, the amount N_{m_n} of the equipment of the device for the comparison of two A_{RNS} numbers is minimal and is defined by the following expression:

$$N_{m_n} = \prod_{k=1}^{n-1} m_k . \quad (3)$$

Variant 2. Let $m_i = m_1 = \min$. In any RNS, we have the minimum possible value of the module $m_i = m_1 = 2$. In this case, the maximum comparison accuracy $W_{m_1} = 1/2$ in the RNS is provided that is determined by the minimum value $m_1 = 2$ of the length of the interval $[jm_1, (j+1)m_1)$. In this variant, the amount of the equipment of the device for arithmetic comparison of two numbers A_{RNS} and B_{RNS} in RNS is maximum, $N_{m_1} = \prod_{k=2}^{n} m_k$.

Since the minimal value of a module for RNS is defined as $m_i = m_1 = 2$, it is obvious that, for both the first and second variants of comparison, the maximum accuracy $W_{\max} = 1$ of comparison of two numbers A_{RNS} and B_{RNS} cannot be reached. Hence, a method for arithmetic comparison should be developed whose result would be determined with the maximum accuracy $W_{\max} = 1$ and, at the same time, a minimum amount $N_{\min}$ of the equipment of the comparing unit would be provided,

i.e., the implementation of the functional $F_{opt.} = W_{\max}(N_{\min})$ should be provided. For this method of arithmetic comparison of two numbers in RNS, which provides the implementation of the functional $F_{opt.}$, two conditions (*requirements*) should be fulfilled. The first (main) condition consists of ensuring the maximum comparison accuracy. In this case, the amount of equipment of the device for comparing two numbers (A_{RNS} and B_{RNS}) will be maximum (formula (2)). The second (secondary) condition consists of providing a minimum amount $N_{\min}$ of the equipment of the comparing device, whenever possible.

In the proposed new method of arithmetic data comparison, the above requirements are implemented as follows. First, the implementation is performed by choosing the maximum RNS base value $m_i = m_n = \max$. In this case, the numerical interval [0, *M*) contains the minimum number of equal numerical intervals $[jm_n, (j+1)m_n)$ (*see Fig. 1*) and thereby provides the fulfillment of the condition of the minimum amount $N_{m_n} = \prod_{k=1}^{n-1} m_k = \min$ of the equipment of the comparison device. Second, the method for comparing two numbers $A_{RNS} = (a_1 \| a_2 \| ... \| a_{i-1} \| a_i \| a_{i+1} \| ... \| a_n)$ and $B_{RNS} = (b_1 \| b_2 \| ... \| b_{i-1} \| b_i \| b_{i+1} \| ... \| b_n)$ that are represented in RNS and belong to one numerical interval $[jm_n, (j+1)m_n)$ must contain an additional procedure of comparison of residues a_n and b_n of these numbers to the maximum RNS module $m_i = m_n = \max$. In this case, the maximum comparison accuracy $W_{\max} = 1$ is reached, namely, right up to unit length intervals, and the functional $F_{opt.}$ reaches its optimum ($W_{\max} = 1$ and $N_{\min} = \prod_{k=1}^{n-1} m_k$). In this case, the residues a_n and b_n are compared simultaneously with the process of formation of SCs.

5 Algorithm for exact arithmetic comparison of two numbers in RNS

When the values of a_n, b_n, n_A and n_B and also the procedure of comparison of two numbers $A_{RNS} = (a_1 \| a_2 \| ... \| a_{i-1} \| a_i \| a_{i+1} \| ... \| a_n)$ and $B_{RNS} = (b_1 \| b_2 \| ... \| b_{i-1} \| b_i \| b_{i+1} \| ... \| b_n)$ are known, the algorithm of exact arithmetic comparison of two numbers in RNS can be represented in the form of the analytical relationships

$$A_{RNS} = B_{RNS}, \text{ if } [(n_A = n_B) \wedge (a_n = b_n)]; \quad (4)$$

$A_{RNS} > B_{RNS}$, if

$$\{(n_A > n_B) \vee [(n_A = n_B) \wedge (a_n > b_n)]\}; \quad (5)$$

$A_{RNS} < B_{RNS}$, if

$$(n_A < n_B) \vee [(n_A = n_B) \wedge (a_n < b_n)]. \quad (6)$$

The algorithm of exact arithmetic comparison of numbers (4)–(6) is used in the method of exact arithmetic comparison of two numbers (A_{RNS} and B_{RNS}) in RNS. The essence of the method is as follows.

1. Represent the numbers $A_{RNS} = (a_1 \| a_2 \| ... \ ... \| a_{i-1} \| a_i \| a_{i+1} \| ... \| a_n)$ and $B_{RNS} = (b_1 \| b_2 \| ... \ ... \| b_{i-1} \| b_i \| b_{i+1} \| ... \| b_n)$ being compared in RNS.

2. Based on the values of a_n and b_n, choose nulevization constants of the form $KH_{m_n}^{(A_{RNS})} = (a_1' \| a_2' \| ... \| a_{i-1}' \| a_i' \| a_{i+1}' \| ... \| a_n)$ and $KH_{m_n}^{(B_{RNS})} = (b_1' \| b_2' \| ... \| b_{i-1}' \| b_i' \| b_{i+1}' \| ... \| b_n)$. Simultaneously compare the values of the residues a_n and b_n of the numbers A_{RNS} and B_{RNS}.

3. Determine the values of A_{m_n} and B_{m_n} multiple of the value of the RNS module m_n as follows:

$$A_{m_n} = A_{RNS} - KH_{m_n}^{(A_{RNS})} = (a_1 \| a_2 \| ... \| a_{i-1} \| a_i \| a_{i+1} \| ... \| a_n) - (a_1^{'} \| a_2^{'} \| ...$$
$$... \| a_{i-1}^{'} \| a_i^{'} \| a_{i+1}^{'} \| ... \| a_n) = [a_1^{(1)} \| a_2^{(1)} \| ... \| a_{i-1}^{(1)} \| a_i^{(1)} \| a_{i+1}^{(1)} \| ... \| 0];$$
$$B_{m_n} = B_{RNS} - KH_{m_n}^{(B_{RNS})} = (b_1 \| b_2 \| ... \| b_{i-1} \| b_i \| b_{i+1} \| ... \| b_n) -$$
$$- (b_1^{'} \| b_2^{'} \| ... \| b_{i-1}^{'} \| b_i^{'} \| b_{i+1}^{'} \| ... \| b_n) = [b_1^{(1)} \| b_2^{(1)} \| ... \| b_{i-1}^{(1)} \| b_i^{(1)} \| b_{i+1}^{(1)} \| ... \| 0].$$

4. Using adders, the collection of constants 0, $m_n, ..., (N-1) \cdot m_n$, and the formulas $A_{m_n} - K_A \cdot m_n = Z_{K_A}^{(A_{RNS})}$ and $B_{m_n} - K_B \cdot m_n = Z_{K_B}^{(B_{RNS})}$, determine the SC components $Z_i^{(A_{RNS})}$ and $Z_j^{(B_{RNS})}$ that are represented in the form $K_{N_{m_n}}^{(n_A)} = \{Z_{N_{m_n}-1}^{(A_{RNS})} \; Z_{N_{m_n}-2}^{(A_{RNS})} \; ... \; Z_2^{(A_{RNS})} Z_1^{(A_{RNS})} \; Z_0^{(A_{RNS})}\}$ и $K_{N_{m_n}}^{(n_B)} = \{Z_{N_{m_n}-1}^{(B_{RNS})} \; Z_{N_{m_n}-2}^{(B_{RNS})} \; ... \; Z_2^{(B_{RNS})} \; Z_1^{(B_{RNS})} \; Z_0^{(B_{RNS})}\}$.

5. Based on the obtained values of the SCs $K_{N_{m_n}}^{(n_A)} = \{Z_{N_{m_n}-1}^{(A_{RNS})} \; Z_{N_{m_n}-2}^{(A_{RNS})} \; ... \; Z_2^{(A_{RNS})} Z_1^{(A_{RNS})} \; Z_0^{(A_{RNS})}\}$ и $K_{N_{m_n}}^{(n_B)} = \; = \{Z_{N_{m_n}-1}^{(B_{RNS})} \; Z_{N_{m_n}-2}^{(B_{RNS})} \; ... \; Z_2^{(B_{RNS})} \; Z_1^{(B_{RNS})} \; Z_0^{(B_{RNS})}\}$, determine the values of the SC bits for which the conditions $Z_{n_A}^{(A_{RNS})} = 0$ and $Z_{n_B}^{(B_{RNS})} = 0$ are satisfied. Find the quantitative values of n_A and n_B of PIPICs.

6. Determine the final result of arithmetic comparison of the numbers A_{m_n} and B_{m_n} according to relationships (4)–(6).

Based on the presented method, a device is developed that implements the process of comparison in RNS, and the patent of Ukraine for this device is acquired [6]. This fact confirms the practical importance of the results of this article.

6 Conclusions of research

In this work, a method is developed for exact arithmetic comparison of data represented in RNS. The method is based on the obtainment and use of PIPICs and maximizes the validity of the result of comparison of numbers in RNS. It is recommended to use it when the operation of arithmetic comparison of data is implemented in hardware in CSPIDs operating in RNSs.

References

[1] I. Ya. Akushskii and D. I. Yuditskii, Machine Arithmetic in Residual Classes [in Russian]: Sov. Radio, Moscow, 1968.

[2] V. A. Krasnobayev, S. A. Koshman, and M. A. Mavrina, “A method for increasing the reliability of verification of data represented in a residue number system”. Cybernetics and Systems Analysis, Vol. 50, Issue 6, pp. 969-976, November 2014.

[3] V. A. Krasnobayev, A. S. Yanko, and S. A. Koshman, “A Method for arithmetic comparison of data represented in a residue number system“. Cybernetics and Systems Analysis, Vol. 52, Issue 1, pp. 145-150, January 2016.

[4] S. A. Moroz and V. A. Krasnobayev, “A data verification method in a non-positional residue number system”. Control, Navigation, and Communication Systems, No. 2 (18), pp. 134-138, 2011.

[5] I. Gorbenko, A., Kuznetsov, M., Lutsenko, and D. Ivanenko, “The research of modern stream ciphers”. 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, pp. 207-210, 2017.

[6] O. Kuznetsov, M. Lutsenko and D. Ivanenko, “Strumok stream cipher: Specification and basic properties”. 2016 Third International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T), Kharkiv, pp. 59-62, 2016.

Рецензент: Сергій Кавун, доктор економічних наук, к.т.н., проф., Харківський технологічний університет "Шаг", вул. Малом'ясницька, 9/11, м. Харків, 61010, Україна.
E-mail: kavserg@gmail.com

Надійшло: Березень 2018.

Автори:

Віктор Краснобаєв, д.т.н., проф., Харківський національний університет імені В. Н. Каразіна, м. Харків, Україна.
E-mail: v.a.krasnobaev@gmail.com

Сергій Кошман, к.т.н., доцент, Харківський національний університет імені В. Н. Каразіна, м. Харків, Україна.
E-mail: s_koshman@ukr.net

Аліна Янко, к.т.н., Полтавський національний технічний університет імені Юрія Кондратюка, м. Полтава, Україна.

E-mail: al9_yanko@ukr.net

Сергій Мороз, к.т.н., Харківський національний технічний університет сільського господарства імені Петра Василенка, м. Харків, Україна.
E-mail: frost9i@ukr.net

Концепція порівняння цілочислових даних, що представлені у системі залишкових класів.

Анотація. Описано методи порівняння цілочислових даних, які представлені у системі залишкових класів (СЗК). Розроблено методику арифметичного порівняння цілочислових даних, яка підвищує точність обробки інформації, представленої у СЗК. Розроблені математична модель і метод точного арифметичного порівняння даних у СЗК, які засновані на отриманні та використанні позиційного ознаки непозиційного коду, забезпечують максимальну достовірність результату порівняння чисел при мінімальній кількості обладнання пристрою, що порівнює. Використання розробленого методу дозволяє підвищити ефективність функціонування спеціалізованих обчислювальних пристроїв у СЗК. На підставі розробленого методу, синтезовано пристрій для реалізації процесу порівняння у СОК, на яке отримано патент України, що підтверджує світову новизну та практичну значимість результатів даної статті.

Ключові слова: система обробки даних; система залишкових класів; цілочислове арифметичне порівняння даних; точність обробки даних; нулевізація числа.

Рецензент: Сергей Кавун, доктор экономических наук, к.т.н., проф., Харьковский технологический университет «ШАГ», ул. Маломясницкая, 9/11, г. Харьков, 61010, Украина.
E-mail: kavserg@gmail.com

Поступила: Март 2018.

Авторы:

Виктор Краснобаев, д.т.н., проф., Харьковский национальный университет имени В. Н. Каразина, г. Харьков, Украина.
E-mail: v.a.krasnobaev@gmail.com

Сергей Кошман, к.т.н., доцент, Харьковский национальный университет имени В. Н. Каразина, г. Харьков, Украина.
E-mail: s_koshman@ukr.net

Алина Янко, к.т.н., Полтавский национальный технический университет имени Юрия Кондратюка, г. Полтава, Украина.
E-mail: al9_yanko@ukr.net

Сергей Мороз, к.т.н., Харьковский национальный технический университет сельского хозяйства имени Петра Василенка, г. Харьков, Украина.
E-mail: frost9i@ukr.net

Концепция сравнения целочисленных данных, представленных в системе остаточных классов.

Аннотация. Описаны методы сравнения целочисленных данных, которые представлены в системе остаточных классов (СОК). Разработана методика арифметического сравнения целочисленных данных, которая повышает точность обработки информации, представленной в СОК. Разработанные математическая модель и метод точного арифметического сравнения данных в СОК, которые основаны на получении и использовании позиционного признака непозиционного кода, обеспечивают максимальную достоверность результата сравнения чисел при минимальном количестве оборудования сравнивающего устройства. Использования разработанного метода позволяет повысить эффективность функционирования специализированных вычислительных устройств в СОК. На основании разработанного метода, синтезировано устройство для реализации процесса сравнения в СОК, на которое получен патент Украины, что подтверждает мировую новизну и практическую значимость результатов данной статьи.

Ключевые слова: система обработки данных; система остаточных классов; целочисленное арифметическое сравнение данных; точность обработки данных; нулевизация числа.

УДК 621.327:621.391

ДВОЙНАЯ ОБФУСКАЦИЯ ТРАНСФОРМАНТ МАЛОРЕСУРСНОГО СТЕГАНОАЛГОРИТМА

Дмитрий Морозов[1], Михаил Шафоростов[1], Сергей Малахов[1], Вадим Сербин[2]

[1] Харьковский национальный университет имени В.Н. Каразина, пл. Свободы, 4, г. Харьков, 61022, Украина
ikurortnik@gmail.com, m61shaforostov@gmail.com, mailgate@meta.ua

[2] ГП «КБ «Южное», ул. Криворожская, 3, 49008, г. Днипро, Украина
buba75@i.ua

Рецензент: Георгий Кучук, д.т.н., проф., НТУ «ХПИ», ул. Кирпичова, 21, г. Харьков, 61000, Украина.
kuchuk56@mail.ru

Поступила в марте 2018

Аннотация. *Целью статьи является ознакомление с основными процедурами адаптивного малоресурсного алгоритма стеганографической обработки изображений и особенностями создания его экспериментальной программы с поддержкой графического интерфейса (мобильного приложения). Предложенная версия программы ориентирована на удобство ее использования на мобильных платформах под управлением операционной системы (ОС) Android. Разработанный адаптивный алгоритм в ручном и автоматическом режимах, позволяет: - определять текущий статус аппаратного обеспечения (мобильной платформы); - учитывать особенности обрабатываемых данных (типы изображений стеганоконтейнера и стеганоконтента); - корректировать параметры работы основных программных модулей исследовательского стеганоалгоритма (блока первичной обработки входных данных (изображений) и блока специальных преобразований – стеганомодуля). Кроме того, проанализированы и другие параметры обработки изображений, имеющих непосредственное влияние на вычислительную сложность всего алгоритма и качество визуализации изображений контейнеров и стеганоконтента. Рассмотренная версия алгоритма является исследовательской и служит средством обеспечения безопасности персональных данных (в данном случае графической информации) пользователей, прежде всего, мобильных гаджетов. Основные свойства синтезированного алгоритма позволяют классифицировать его, как программное средство обеспечения стеганографической защиты, локализованное для условий внутрикадровой обработки изображений. Представленная версия алгоритма требует его дальнейшего совершенствования и имеет своей главной целью подтвердить правильность выбранных методов обработки данных и стратегии создания пользовательского интерфейса соответствующего мобильного приложения.*

Ключевые слова*: обфускация; кодирование с преобразованием; зональное кодирование; стеганография.*

1 Введение

Известно, что одним из эффективных направлений обеспечения сокрытия факта передачи и хранения информации, является применение различных стеганографических методов [1-3]. В рамках данной работы рассматривается цифровое направление стеганографии, локализованное в области изучения возможностей синтеза малоресурсного алгоритма инкапсуляции статических цифровых полутоновых изображений (далее *стеганоконтента*) в другое статическое цифровое изображение (далее *стеганоконтейнер*).

Сужение области исследований в направлении создания легковесного стеганоалгоритма обусловлено практическим интересом создания соответствующего программного решения с дружественным интерфейсом, ориентированного для использования на различных мобильных устройствах (смартфоны, планшеты и т.п.). Предварительно, такой алгоритм должен обеспечивать адаптацию параметров своей работы к параметрам работы и характеристикам мобильной платформы (в «*автоматическом*» режиме работы) или уведомлять о своих возможностях (в «*интерактивном*» режиме) при обеспечении требуемых характеристик формируемого стеганоконтента в сложившихся условиях ресурсных ограничений гаджета (количество и вычислительная сложность работающих активных приложений/задач, текущая температура, фактический заряд встроенной аккумуляторной батареи (АБ) и т.п.). Под уведомлением своих возможностей следует понимать процесс информирования пользователя гаджета о расчетных параметрах его работы (объем обрабатываемого контента или время обработки) для текущих характеристик обрабатываемых изображений (разрешение изображений и их

количество) при требуемом уровне обеспечиваемой скрытности инкапсулируемого контента.

Снижение общей вычислительной сложности стеганоалгоритма (*особенно при пакетной обработке данных*) имеет своей целью обеспечение условий для его последующего применения в составе программного обеспечения различных мобильных платформ (гаджетов), с присущими им качествами и особенностями: 1 – постоянное изменение текущего уровня заряда встроенной АБ, в том числе изменение паспортных характеристик работы АБ по причине ее износа (старения); 2 - поддержка режима многозадачности при работе с мобильными приложениями; 3 - неравномерность в распределении имеющихся вычислительных ресурсов при относительном «равенстве» доступа к ресурсу бортовой АБ между:

а) используемыми мобильными приложениями *(активными приложениями и задачами выполняемых в фоновом режиме (с приоритетом в обслуживании сервисов реального времени))*;

б) имеющимися аппаратными модулями самого гаджета *(например, модулями беспроводной связи (Wi-Fi или Bluetooth), активация функций «работа с гарнитурой» или «фонарик» и т.п.)*;

4 - изменение параметров функционирования отдельных аппаратных элементов мобильной платформы (например, излучаемой мощности *Wi-Fi* модуля) и активных приложений при достижении контрольных (*предустановленных*) значений разряда АБ.

Применительно к очерченной предметной области, задачу можно сформулировать так: – какой объем стеганоконтента с различной степенью его стойкости к обнаружению и декодированию можно сформировать при разных параметрах/условиях работы гаджета (*тип запущенных приложений, текущее состояние бортовой АБ и т.п.*). Цели таких изысканий очевидны – это автоматическое формирование рекомендованных значений параметров работы стеганоалгоритма (*особенно в режиме пакетной обработки данных*) в зависимости от текущих характеристик работы гаджета или же изменение параметров работы самого устройства для достижения требуемых характеристик (*качества*) формируемого стеганоконтента.

2 Основная часть

Сокрытие в цифровых объектах (в данном случае, полутоновых изображениях) какой-либо дополнительной информации, вызывает некоторые искажения этих объектов-контейнеров [1]. При сбалансированных настройках соответствующего алгоритма обработки, возникающие искажения находятся ниже порога чувствительности среднестатистического человека, и не приводят к визуально заметным изменениям/искажениям этих объектов. Таким образом обеспечивается требуемый баланс между сохранение типовых характеристик используемого графического формата представления данных, а так же количеством и интенсивностью проявления различных артефактов изображений, дающих лишний повод задуматься, собственно, о причине их появления. Однако, в оцифрованных объектах, изначально имеющих аналоговую природу, всегда присутствует шум квантования, а при воспроизведении этих объектов проявляются нелинейные искажения, обусловленные особенностями аппаратной части, что способствует дополнительной маскировке сокрытой информации [2].

В основе рассматриваемого алгоритма сокрытия информации лежит внутрикадровый алгоритм сжатия изображений, основанный на использовании метода кодирования с преобразованием (дискретное косинусное преобразование - ДКП) [4,5]. Использование свойств ДКП позволяет встраивать стеганоконтент (полутоновое изображение - 256 градаций яркости) в матрицу коэффициентов преобразования [6] изображения-контейнера. Процедура инкапсуляции стеганоконтента (коэффициентов преобразования скрываемого изображения), реализуется после проведения селекции коэффициентов преобразования изображения-контейнера [4,7]. Для восстановления исходных изображений выполняется обратное ДКП (ОДКП) [5].

Использование методов кодирования с преобразованием обеспечивает получение матриц спектральных коэффициентов (трансформант), в которых большая часть коэффициентов либо близки, либо равны 0. Кроме того, учет свойств зрения человека [5], позволяет аппроксимировать коэффициенты без заметной потери качества визуализируемых изображений. Для

этого используется специализированный механизм квантования коэффициентов [3].

Следует отметить, что в данном случае использовалась симметричная схема реализации кодирования изображений контейнера и контента, т.е. в обоих случаях размер субблоков изображений был одинаковой размерности. Это несколько сужало диапазон возможных настроечных вариаций стеганоалгоритма, но не меняло сути наблюдаемых процессов. Для описания сути алгоритма достаточно удобно рассматривать его отдельными этапами:

1-й этап – деление на блоки и сглаживание исходных изображений; 2-й этап – формирование серий подобных (*идентичных по содержанию*) блоков исходных изображений; 3-й этап – проведение ДКП (для изображений контейнера и стеганоконтента); 4-й этап – проведение модифицированного зонального отбора коэффициентов [7]; 5-й этап – встраивание и обфускация (внутриблочная и межблочная) коэффициентов ДКП; 6-й этап – результирующая обработка и формирование массива «сжатого» стеганокадра.

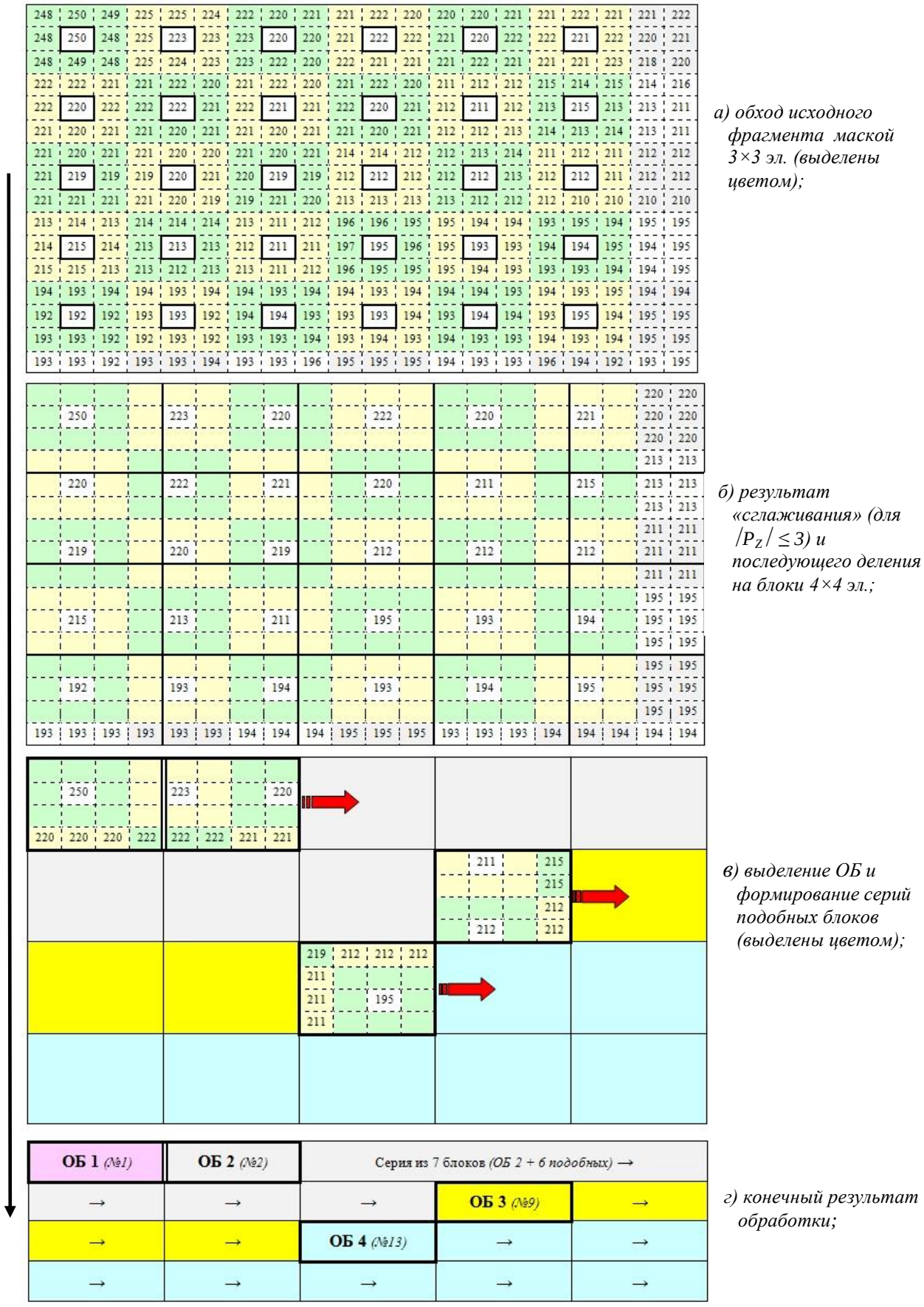
Основной целью 1-го этапа является уменьшение количества визуально не фиксируемых перепадов яркости элементов исходных изображений [5]. Для этого изображения разбиваются на блоки размером 3×3 элемента, после чего в каждом блоке оценивается разница значений центрального элемента с остальными. В случае если полученная разница меньше установленного значения порога загрубления (P_Z), то элемент (пиксел) в этой позиции замещается значением яркости центрального элемента (см. рис.1а). В результате, получаем новые, сглаженные изображения (рис.1б), учитывающие особенности локального распределения яркости элементов исходных изображений для маски 3×3 эл. Этот прием позволяет уменьшить вычислительную сложность всего алгоритма на последующих этапах его работы.

На втором этапе скрываемое изображение делится на блоки установленного размера и в каждом из них определяются элементы с максимальным и минимальным значениями яркости. Далее в соседних блоках изображений (при построчном обходе) эти значения последовательно анализируются и если разности максимальных и минимальных элементов меньше установленного порога загрубления (P_Z), то такие блоки считаются идентичными (*или подобными*), а первый такой блок в серии - опорным блоком (ОБ) (рис.1 в). В результате получаем массив, где указывается порядковый номер каждого ОБ и, соответственно, число его повторов в каждой новой серии (рис.1г). Проведение перечисленных процедур обуславливает необходимость проведения кодирования с преобразованием (в нашем случае ДКП) только для ОБ. Таким образом удается уменьшить общую вычислительную сложность всего алгоритма (*уменьшив его самую ресурсоемкую часть*).

Третий этап представляет собой проведение ДКП над каждым блоком изображения контейнера и ОБ скрываемого контента.

На следующем (4-ом) этапе реализуются процедуры модифицированной зональной селекции коэффициентов преобразования. Вариант маски зонального отбора коэффициентов для блоков размером 8×8 элементов представлен ниже (рис. 2а и 3а). Применение зонального способа отбора исключает необходимость адресации знакомест сохраняемых коэффициентов [4,7] и экономит ресурсы памяти мобильного устройства. Предварительно, для изображения-контейнера сохраняется 10 коэффициентов (рис. 2а), а для улучшения качества восстановления встраиваемых изображений сохраняется несколько большее количество (*в данном случае 14*) коэффициентов преобразования (рис. 3а).

На 5-м этапе производится обфускация коэффициентов ДКП для всех блоков изображений контейнера и контента. Данная процедура выполняется в два этапа с использованием любой из возможных масок обфускации (рис.4,5). В тестовой версии алгоритма соответствующие пары масок перемешивания (для контейнера и контента) строго взаимосвязаны, а их общее количество зависит от размерности используемых блоков. Поскольку позиции коэффициентов не накладываются друг на друга, то полученные после перемешивания массивы совмещаются в одной матрице (рис.4). Размерность субблоков может задаваться вручную и автоматически (*зависит от режима работы алгоритма*), а используемая маска перестановок формируется автоматически, случайным образом. Эти параметры указываются в соответствующей позиции формируемого сжатого файла стеганоконтейнера.



а) обход исходного фрагмента маской 3×3 эл. (выделены цветом);

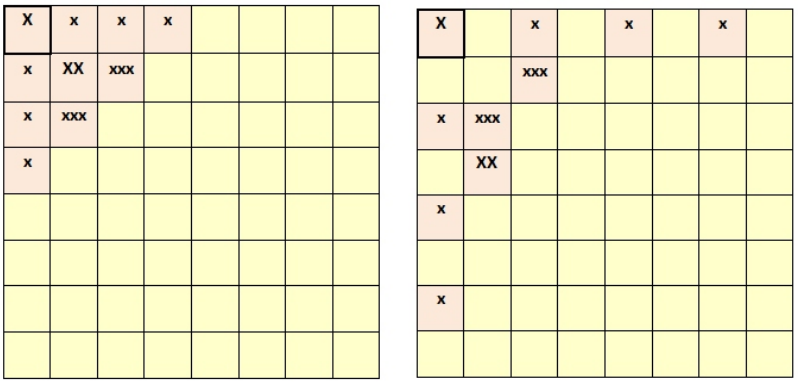
б) результат «сглаживания» (для $|P_Z| \le 3$) и последующего деления на блоки 4×4 эл.;

в) выделение ОБ и формирование серий подобных блоков (выделены цветом);

г) конечный результат обработки;

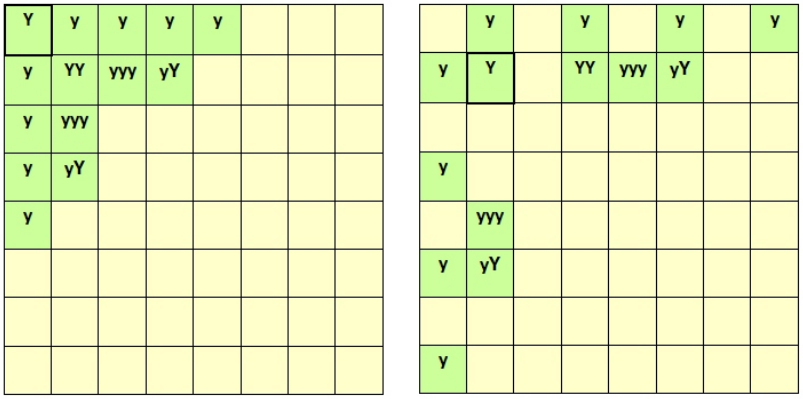
Рис.1 - Формирование серий подобных блоков
(*для блоков размером 4×4 эл.*)

Очевидно, что возможное количество перестановок зависит от размера используемых блоков, поэтому при необходимости увеличить степень скрытности контента (*при неизменном количестве сохраняемых коэффициентов преобразования*), размер блоков увеличивается (*проводится по результатам оценки текущего состояния ресурсов гаджета*).



а) *б)*

Рис. 2 - Маска зонального отбора (*а*) и вариант размещения (*б*) коэффициентов «контейнера»



а) *б)*

Рис. 3 - Маска зонального отбора (*а*) и вариант размещения (*б*) коэффициентов «контента»

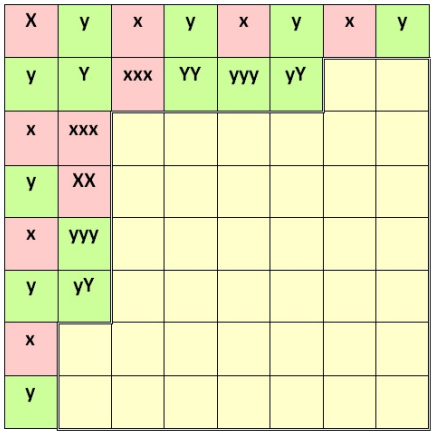


Рис. 4 - Размещения коэффициентов в стеганоконтейнере (*вариант*)

Более того, в ходе экспериментов была проверена идея несимметричной внутриблочной обработки (*размеры субблоков контейнера всегда больше блоков контента*), использование которой значительно затрудняет локализацию элементов инкапсулированного контента и расширяет диапазон возможных перестроек алгоритма. Однако, данный режим обработки несколько сложнее в исполнении и требует введения дополнительных служебных маркеров, позволяющих автоматизировать безошибочное извлечение контента при его декодировании. Результаты такой обработки будут представлены несколько позже.

Для увеличения сложности процедур поиска и извлечения скрытого контента, в исследовательской версии алгоритма, применялся упрощенный механизм межблочной обфускации (*только перемешивание ОБ изображения контейнера* (рис.5)). При этом блоки стеганоконтента встраиваются не последовательно, а в соответствии с генерируемой маской межблочного перемешивания, являющейся еще одним элементом ключа, необходимым для извлечения и последующего декодирования скрытой информации.

Как и при внутриблочной обработке (обфускации), информация о номере использованного варианта маски межблочного перемешивания, указывается в соответствующей позиции формируемого файла стеганоконтейнера. Очевидно, что использование двухуровневой обфускации стеганоконтента ощутимо расширяет возможности самого алгоритма.

В результате проведения межблочной обфускации формируется первичный стеганокадр. Однако для него присущ один существенный недостаток. После формирования серий подоб-

ных блоков, количество ОБ всегда будет меньше количества блоков изображения-контейнера. Как следствие, может возникнуть ситуация, когда не все блоки контейнера будут «заполнены» информацией стеганоконтента. Так в «рабочих» матрицах - контейнерах предварительно сохраняется 24 коэффициента (рис.4), а в «пустой» матрице только 10 (рис. 2). Это обстоятельство приведет к тому, что при декодировании изображений трудно организовать работу счетчика блоков (матриц-контейнеров). Кроме того, упрощается задача анализа содержимого всего стеганокадра и, как следствие, выявление «заполненных» блоков со всеми вытекающими отсюда последствиями.

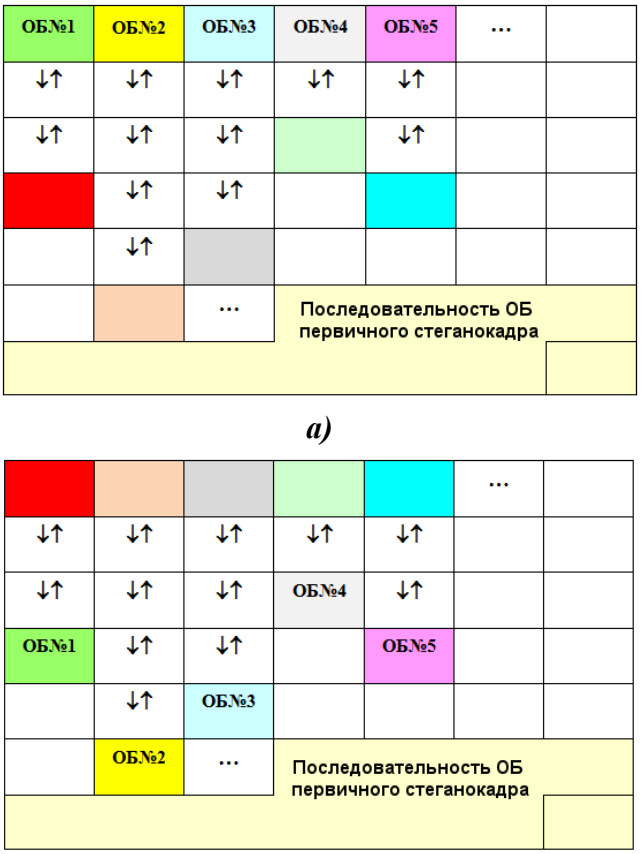


Рис. 5 - Исходное (а) и конечное (б) положение матриц-контейнеров до и после проведения межблочной обфускации (*вариант маски*)

Для устранения указанных недостатков необходимо «выровнять» объем цифрового описания всех блока первичного стеганокадра. С этой целью обеспечивается заполнение отсутствующих позиций (в текущем варианте используемой маски) дополнительным балластным содержимым. Для заполнения отсутствующих позиций в тестовой версии алгоритма использовать «родные» значения коэффициентов преобразования, полученные для изображения контейнера. Такой подход имеет двойной эффект: - затрудняет обнаружения аналитиком скрытой информации; - улучшает параметры восстановления исходного изображения контейнера для данной группы блоков.

Последний этап алгоритма предполагает проведение ряда технологических процедур (*в т.ч. характерных и для традиционной реализации алгоритма сжатия JPEG [3,5,8,9]*), таких как уменьшение разрядности и кодирование полученных значений, а также формирование служебного заголовка (ключа) «сжатого» стеганокадра.

Для уменьшения разрядности значений первичного стеганокадра выполняется нормировка значений каждого сохраняемого блока. Исходя из того, что значения яркости исходных изображений варьируются в рамках от 0 до 255, то нормировка значений производится относительно среднего их значения (127) с округлением результата до ближайшего целого.

Учитывая, что все блоки формируемого стеганокадра имеют одинаковый диапазон изменения значений яркости элементов после их нормировки, то обеспечивается возможность синтезировать общий «словарь» для всего стеганокадра. Поэтому дальнейшая кодировка значений первичного стеганокадра обеспечивается применением методов кодирования без потерь информации [5,8,9], например: - методом длин серий или методом Хаффмана.

3 Результаты работы алгоритма

В ходе проведения экспериментов исследовался характер влияния определенных параметров алгоритма на характеристики визуализации изображений и параметров встраивания скрытого контента. Среди таких параметров были использованы следующие: - количество ОБ; - тип изображений контейнера и контента; - размер субблоков изображений (8,16,32); - величина порога загрубления, P_Z (от 1 до 20); - уровень заряда АБ гаджета.

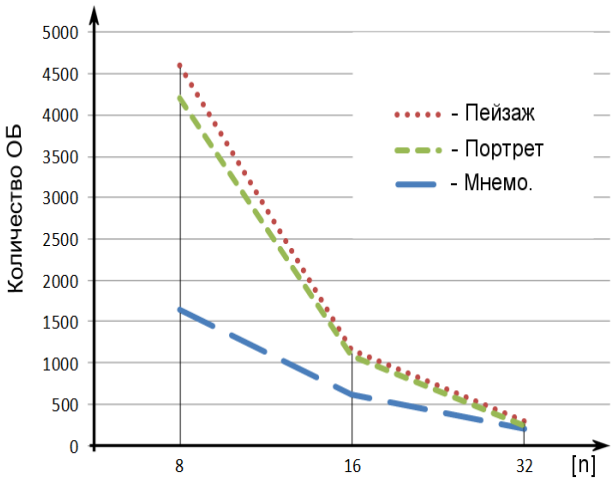
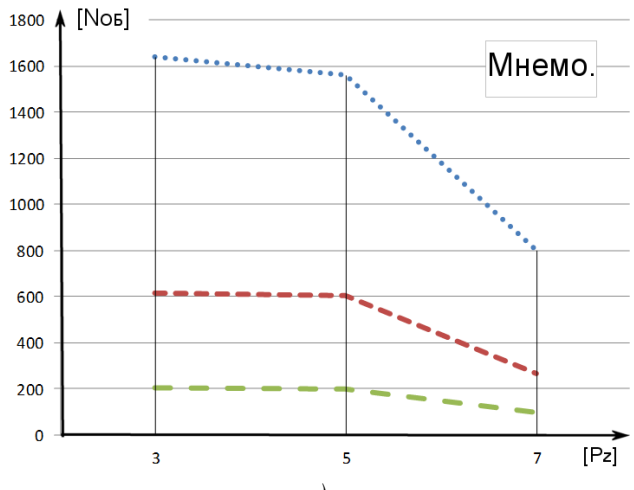


Рис. 6 – Зависимость количества ОБ от n для разных типов изображений

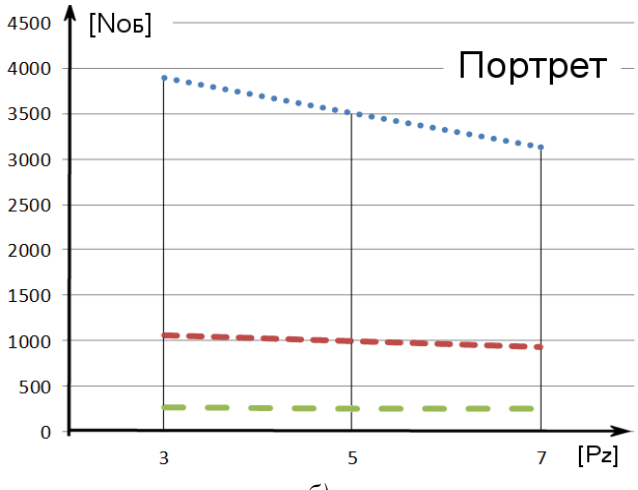
В общем случае количество формируемых ОБ зависит от установленного значения порога загрубления (P_Z). С учетом особенностей восприятия полутоновых изображений [5], характеризующихся различной вероятностью перепада яркости между соседними элементами *(изображения типа мнемосхема, портрет и пейзаж (аэрофотоснимок))*, оптимальный варианты изменения величины P_Z следует ограничить значениями 3,5,7. При таких значениях удается получить наиболее приемлемый баланс между визуальной заметностью проявляющихся искажений и снижением вычислительной сложности алгоритма.

Так на рис. 6 представлены зависимости, отражающие взаимосвязь количества формируемых ОБ с размерностью используемых субблоков (n×n), которые используются при обработке изображений контейнера и контента.

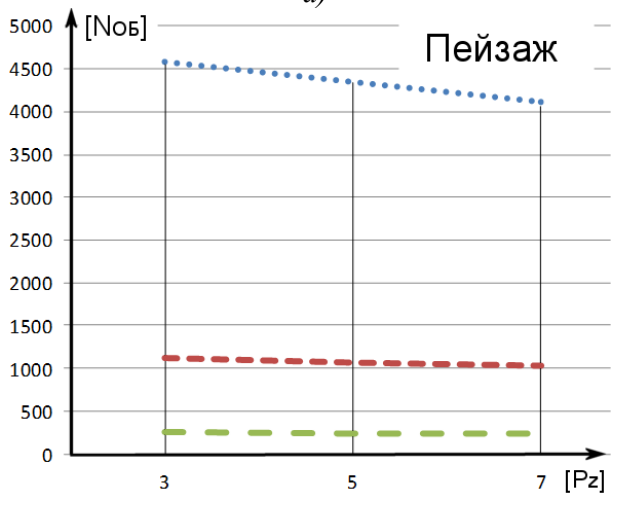
Характер зависимости количества формируемых ОБ ($N_{ОБ}$) от введенного значения P_Z для разных типов изображений (различной вероятности (p) перепада яркости) отображает рис.7.



а)



б)



в)

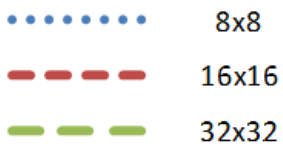
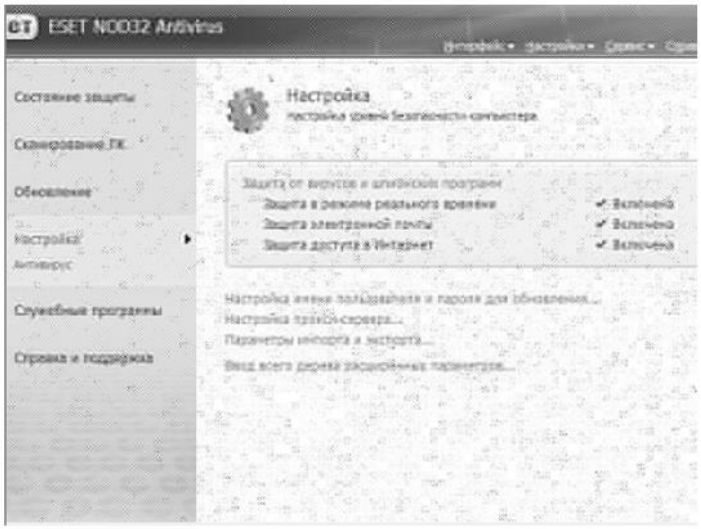


Рис. 7 – Зависимость количества ОБ от P_Z при изменении размерности блоков для различных типов изображений:
а) – мнемосхема *(0,01 ≤ p ≤ 0,03)*;
б) – портрет *(0,03 ≤ p ≤ 0,06)*;
в) – пейзаж/аэрофото *(0,06 ≤ p ≤ 0,1)*.

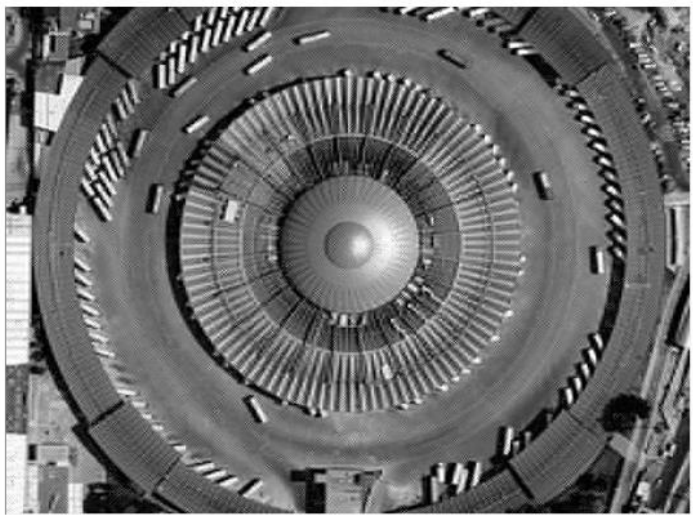
Из анализа представленных результатов следует, что количество формируемых ОБ зависит от 3-х параметров:

- размера субблоков; значения порога загрубления (P_Z); - типа обрабатываемого изображения (*изображений с разной сложностью структуры, т.е. разной вероятностью перепада яркости соседних пикселов*). Так, при увеличении размера субблоков, уменьшается их общее количество в кадре и, соответственно, уменьшается количество ОБ. При увеличении значения P_Z (*при лучших вариантах обработки 3,5,7*) расширяется соответствующий диапазон подобия, поэтому общее количество ОБ уменьшается.

Данные изображения-контента сохраняются в виде значений ОБ, потому от количества инкапсулированных ОБ зависит интенсивность искажений изображения-контейнера. В качестве примера на рис. 8 представлен результат вставки в изображения с разной структурой (*мнемосхема (а) и аэрофотоснимок (б)*) одного и того же изображения (*одинаковое количество ОБ*) с высокой вероятностью перепада яркости соседних элементов (*типа пейзаж*).



а) фрагмент рабочего окна одной из программ;



б) аэрофотоснимок автовокзала в Мехико.

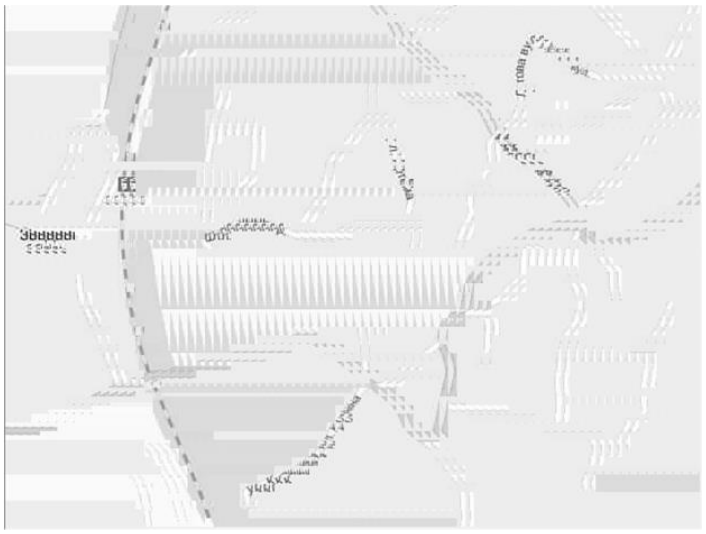
Рис. 8 – Результаты стегановставки
(n=8; P_Z *=3; ОБ 4608)*
в контейнеры с разной вероятностью перепада яркости соседних элементов

Как следует из приведенного примера, при увеличении количества ОБ, в процессе вставки контента в контейнер с простой структурой, визуально увеличиваются искажения изображения-контейнера (*особенно на однородных областях с низкой детальностью фрагментов*), что демаскирует результаты инкапсуляции стеганоконтента. Напротив, результат подобной вставки в изображения со сложной структурой, практически не демаскируется.

Очевидно, что при инкапсуляции изображений с различным количеством ОБ, в изображение контейнер со сложной структурой (*большой вероятностью перепада яркости*), интенсивность визуально фиксируемых искажений контейнера крайне незначительна.

По результатам экспериментального моделирования следует отметить следующее: 1 - при стегановставке изображений с более сложной структурой, чем сам контейнер, наблюдаются интенсивные артефакты, в отличие от случаев инкапсуляции контента в сложные изображения, когда проявляющиеся искажения могут быть практически незаметны; 2 - независимо от размера используемых субблоков (*смоделировано для случаев 8×8, 16×16, 32×32*), изображения-контейнеры со сложной структурой, с точки зрения стеганографических преобразований, всегда имеют неизменно больший потенциал (*лучшая способность к пространственному маскированию визуально фиксируемых артефактов, соответственно и низкая демаскировка факта выполненной стегановставки*).

Проведенные эксперименты подтвердили серьезное влияние значения порога загрубления на «заполнение» контейнера и качество восстановление самого контента (характерный пример представлен на рис. 9).



а) Pz = 20

б) Pz = 1

Рис. 9 – Восстановленный контент
с «простой» структурой *(типа мнемосхема)*
для разных значений P_Z

Так, при увеличении P_Z результирующий коэффициент сжатия будет большим (*т.к. формируется меньшее количество ОБ*) и уменьшается общее время кодирования (*время обработки стеганоконтента*), и наоборот, при уменьшении значения P_Z формируется больше ОБ и увеличивается время обработки. Однако, увеличение значения P_Z влияет на качество восстановления инкапсулированных изображений в худшую сторону (динамика процесса хорошо видна на рис.9), т.к. наблюдается результат работы блока ускорителя (*формирование подобных блоков*). И напротив, при уменьшении P_Z возрастает количество ОБ, что позитивно скажется на характеристиках визуализации изображения-контента, однако при этом увеличится количество артефактов в изображении-контейнере.

Другими словами, в процессе определения требуемого значения P_Z необходимо соблюдение баланса между качеством (*степенью сжатия*) визуализации скрываемого контента и интенсивностью, и характером визуализации артефактов, демаскирующих процесс выполненной стегановставки в изображение – контейнер.

Как следует из результатов моделирования, существенным фактором, влияющим на природу и интенсивность фиксируемых искажений изображений-контента, является характер использования/расстановки ОБ. Так, даже при обработке изображений с относительно простой структурой могут возникнуть довольно заметные искажения при его восстановлении (*дорожки повторяющихся блоков* на рис. 9). Ошибки вызваны тем, что сравниваются только max и min значения элементов в соседних блоках (*безотносительно позиции их размещения*). Поэтому при сравнении содержания соседних блоков, фактические значения могут не выходить за указанные пределы P_Z , и как следствие, алгоритмом формируются целые серии верных по содержанию, но ошибочных по структуре ОБ.

Упрощенный пример такой ситуации представлен на рис.10. В данном случае имеем фрагмент изображения (рис.4а), содержащий последовательность из 4-х блоков с одинаковыми значения яркости пикселов, но их различным размещением в каждом из блоков.

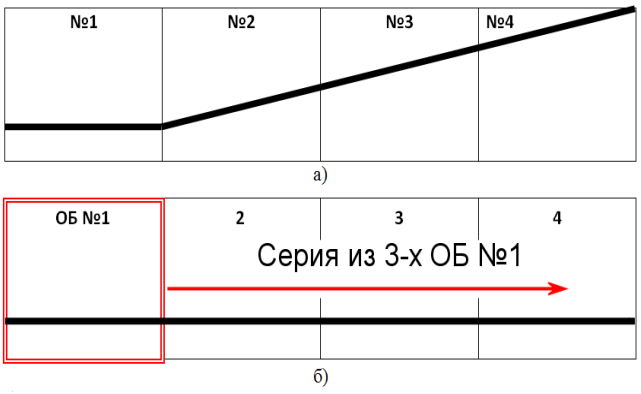


Рис. 10 – Результат формирования ОБ
(по старой схеме)

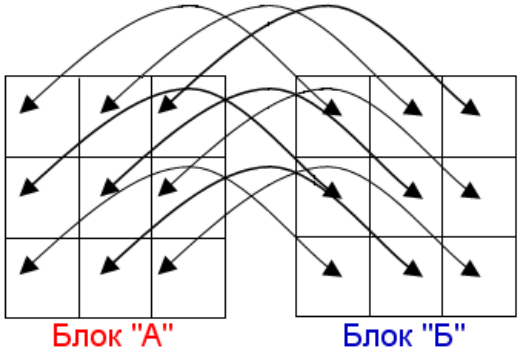


Рис. 11 – Поэлементное сравнение
(усовершенствованная схема)

Так в 1-м блоке есть горизонтальная линия, которая в начале 2-го блока меняет свое направление (поднимается). В результате работы алгоритма по «старой» схеме (рис. 1), при формировании серий ОБ, разница max и min значений перепада яркости элементов в соседних блоках будет неизменной и находиться в границах заданного значения P_Z. Поэтому при формировании серии ОБ, «родные» значения элементов в блоках №№ 2,3,4, естественно, не сохраняются (рис.10 б). Последствия такой обработки видны на рис. 9.

Для устранения указанного недостатка, используемый механизм был несколько доработан (рис.11). В его усовершенствованной версии производится позиционное сравнение элементов в соседних блоках.

Пример результата применения «старой» и доработанной схемы обработки представлен на рис.12 (*для контента с простой структурой (типа мнемосхема)*).

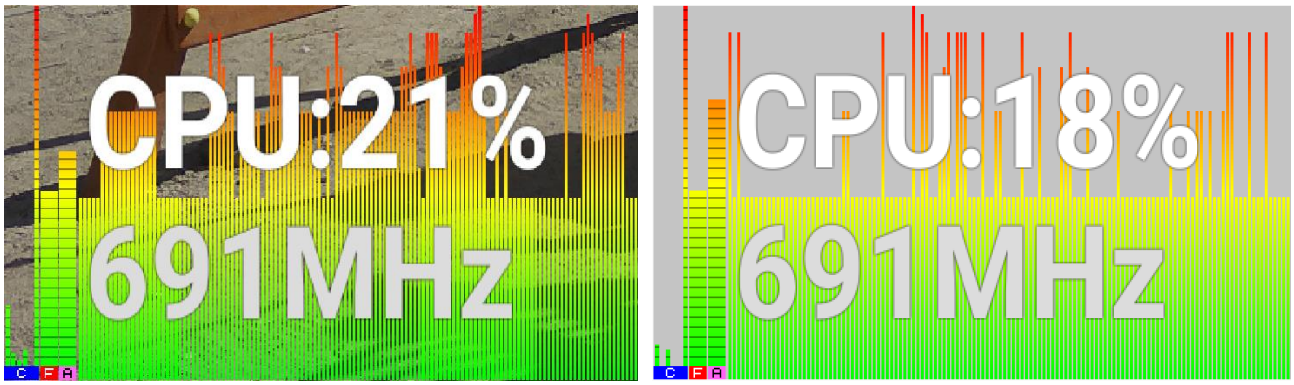
Из приведенного примера следует, что указанная доработка позволила ощутимо улучшить параметры восстановления скрываемого контента при относительно небольшом изменении количества ОБ (*мене чем на четверть*).

Тестовые испытания алгоритма проводилось на смартфоне под управлением ОС Android v. 6.0.1 Marshmallow® (*для данной версии ОС Android® актуальна версия Android SDK API v23*). Адаптация программного кода алгоритма велась с использованием среды разработки Intellij IDEA 2017.1.1.

а) n=8; P_Z =3; ОБ 1435; *б) n=8; P_Z =3; ОБ 1708*

Рис. 12 – Результаты восстановления контента для старой (а) и новой (б) схем формирования ОБ

Исследования тестовой версии алгоритма проводились с параметрами разрешения экрана смартфона 1080×1920 и частотой обновления 60Hz.



а) *б)*

Рис. 13 – Результаты измерений до запуска алгоритма (а) и в ходе его работы (б), где:
C – загрузка процессора; F – частота процессора; A – потребление АБ.
Прим.- указатели индикаторов размещаются в левой нижней части экрана.

Рис. 14 – Пользовательский интерфейс приложения (*тест-версия*)

Для проведения измерений использовались специализированные мобильные приложения (*например, «Advanced Task Manager», «Cool Tool», «OS Monitor» и др.*), с помощью которых фиксировались соответствующие данные: - объем занятой оперативной памяти, состояние АБ, загрузка процессора и его тактовая частота. Оценка объемов занятой оперативной памяти выполнялась для субблоков различных размеров, при разных уровнях заряда бортовой АБ и в отсутствии выполнения сторонних приложений (*за исключением системных процессов*). Характерные показания измерений при работе запущенного тестового алгоритма представлены на рис. 13 *(для варианта с размером субблоков изображений 8×8 эл.)*.

На рис. 14 представлен вариант исполнения основной панели управления приложения. Графический интерфейс программной оболочки, реализующей тестовую версию алгоритма, имеет несколько функциональных разделов, обеспечивающих возможность независимого управления основными процедурами алгоритма (*рабочие папки, режимы выбора масок обфускации, размеры блоков, порог сглаживания, параметры симметрии алгоритма и др.*).

В результате проведения цикла экспериментов, удалось установить, что при использовании адаптивного варианта работы алгоритма (*без вмешательства человека в порядок формирования настроечных параметров алгоритма*), увеличивается общее время использования ресурса АБ мобильного устройства (*особенно в режиме пакетной обработки данных*). Т.е. таким образом удалось подтвердить правильность выбранной концепции создания алгоритма: - согласование основных параметров работы алгоритма с текущими характеристиками и условиями работы мобильной платформы.

На рис. 15 представлены типовые значения загрузки процессора мобильного устройства (*обычно в диапазоне 12÷18 %*), характерные для случая выполнения полного цикла стеганографической обработки изображений (*без промежуточных стопов и формирования служебных данных телеметрии*).

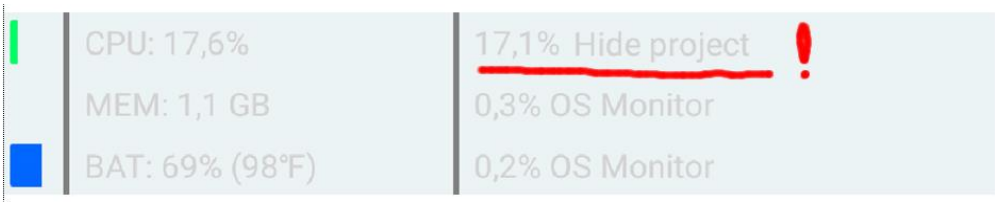


Рис. 15 – Результат измерения загрузки процессора при выполнении алгоритма

4 Выводы

1. Рассмотренный алгоритм обеспечивает режим адаптивной малоресурсной инкапсуляции стеганоконтента различного типа в изображения – контейнеры соизмеримых размеров (*количество блоков контейнера всегда больше количества формируемых блоков контента*).

2. При встраивании и декодировании (*извлечении*) контента, ключевой информацией являются следующие параметры: - использованный размер субблоков; - параметр симметрии стегановставки; - идентификаторы масок внутриблочной и межблочной обфускации.

3. Предложенный алгоритм обеспечивает адаптацию параметров своей работы к типу и количеству (*пакетная обработка*) обрабатываемых данных с учетом текущих характеристик функционирования мобильной платформы. Этим обеспечивается согласование параметров работы алгоритма с текущей загрузкой и ресурсными возможностями устройства.

4. Представленный порядок обработки трансформант характеризуется простотой реализации, малой вычислительной сложностью и обеспечивает хорошие условия для последующей инкапсуляции стеганоконтента с использованием симметричной схемы стегановставки.

5. Для повышения скрытности и стойкости к несанкционированной экстракции стеганоконтента использован гибридный механизм обфускации, реализующий механизмы внутриблочного и межблочного (внутрикадрового) перемешивания.

6. Сделан вывод о необходимости автоматизации процедуры выбора типа контейнера, в зависимости от характеристик инкапсулируемых данных.

7. Совершенствование рассмотренного алгоритма видится по следующим основным направлениям: – разработка эффективного механизма формирования масок межблочной обфускации; – развитие направления несимметричной обработки контейнера и контента; – комбинирование различных видов преобразований, в зависимости от типа обрабатываемого контента; – разработка механизма защиты от размножения ошибок стегановставки; – оптимизация параметров работы модуля «ускорителя» в зависимости от типа изображения – стеганоконтента; – формирование алгоритма скрытой трансляции видеоряда.

Ссылки

1. Gribunin, V.G. Tsifrovaya steganografiya / Gribunin V. G., Okov I. N., Turintsev I. V. – M.: Solon-Press, 2002. – 272 p.
2. Konakhovich, G.F. Komp'yuternaya steganografiya. Teoriya i praktika / Konakhovich G. F., Puzyrenko A.Yu. – K.: MK-Press, 2006. – 288 p.
3. Bykov, S.F. Algoritm szhatiya JPEG s pozitsii komp'yuternoi steganografii / Bykov S. F. // Zashchita informatsii. Konfident. – SPb.: 2000, № 3. pp. 26.
4. Prett, U. Tsifrovaya obrabotka izobrazhenii / U. Prett. – M.: Mir, 1985. – 736 p.
5. Zubarev, Yu. B. Tsifrovaya obrabotka televizionnykh i komp'yuternykh izobrazhenii / Yu.B. Zubarev, V.P. Dvorkovich. – Moskva: MTsNTI, 1997. – 212 p.
6. Korolev, A.V. Otsenka informativnosti transformant diskretnogo kosinusnogo preobrazovaniya / A.V. Korolev // Sistemi obrobki informatsiї. – 2003. – Vip.3. pp.81–85.
7. Malakhov, S.V., Bukhantsov, A.D. Zonal'noe kodirovanie izobrazhenii s razlichnym razbieniem prostranstvenno-chastotnoi oblasti / S.V. Malakhov, A.D. Bukhantsov // Sistemi obrobki informatsiї. – 2001. – Vip. 4(14). pp. 121–125.
8. Mastryukov, D. Algoritmy szhatiya informatsii. Ch.1. Szhatie po Khaffmenu // Monitor. - 1993. - № 7-8. pp.14-20.
9. Mastryukov, D. Algoritmy szhatiya informatsii. Ch.7. Szhatie graficheskoi informatsii // Monitor. - 1994. - № 6. pp.12-20.

Reviewer: Georgiy Kuchuk, Doctor of Technical Sciences, Full Professor, Professor of the Department of Computer Science and Programming, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine.
E-mail: kuchuk56@ukr.net

Received: March 2018.

Authors:
Dmitriy Morozov, student of CSD, V. N. Karazin Kharkiv National University, Kharkov, Ukraine. E-mail: ikurortnik@gmail.com
Mykhailo Shaforostov, student of CSD, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
E-mail: m61shaforostov@gmail.com
Serghii Malakhov, Ph.D., Senior Researcher, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
E-mail: mailgate@meta.ua
Vadim Serbin, Leading Specialist, Yuzhnoye State Design Office, Dnipro, Ukraine. E-mail: buba75@i.ua

The double obfuscation of transformants of low-resource steganography algorithm.

Abstract. The purpose of the article is to familiarize with the basic procedures of the adaptive low-resource algorithm of steganography processing of images and the peculiarities of creation of its experimental program with support of the graphic interface (mobile application). The version of the program offered in operation provides convenience of its use on mobile platforms under control of the Android® operating system (OS). The adaptive algorithm in the manual and automatic modes is developed allows to define: - a current status of the hardware (a mobile platform); - take into account the features of processed data (types of images to a steganocontainer and to a steganocontent); - to adjust parameters of operation of main software modules of a research steganoalgorithm (the modul of preprocessing of input data (images), and the module of special conversions - the steganographic module). In addition, other parameters were investigated of image processing having direct influence on computing complexity of all algorithm and quality of visualization images of containers and steganocontent. Presented version of the algorithm is the research version and is an instrument for ensuring of the security of personal information (in this case the graphic information) users, first of all, of mobile gadgets. The main properties of the synthesized algorithm, allow classifying it as software of steganography protection, localized for conditions intraframe processed of images. Presented version of the algorithm requires its subsequent enhancement and has the main goal to confirm correctness of the selected methods of data handling and strategies of creation of the user interface for corresponding mobile application.

Keywords: obfuscation; encoding with conversion; zonal encoding; steganography.

Рецензент: Георгій Кучук, д.т.н., проф., НТУ «ХПІ», м. Харків, Україна.
E-mail: kuchuk56@ukr.net

Надійшло: Березень 2018.

Автори:
Дмитро Морозов, студент ФКН, ХНУ імені В. Н. Каразіна, м. Харків, Україна. E-mail: ikurortnik@gmail.com
Михайло Шафоростов, студент ФКН, ХНУ імені В. Н. Каразіна, м. Харків, Україна. E-mail: m61shaforostov@gmail.com
Сергій Малахов, к.т.н., с.н.с., ХНУ імені В. Н. Каразіна, м. Харків, Україна. E-mail: mailgate@meta.ua
Вадим Сербін, провідний фахівець, ДП «Конструкторське бюро «Південне», м. Дніпро, Україна. E-mail: buba75@i.ua

Подвійна обфускація трансформант малоресурсного стеганоалгоритма.

Анотація. Метою статті є ознайомлення з основними процедурами адаптивного малоресурсного алгоритму стеганографічної обробки зображень і особливостями створення його експериментальної програми з підтримкою графічного інтерфейсу (мобільного додатка). Запропонована версія програми орієнтована на зручність її використання на мобільних платформах під управлінням операційної системи (ОС) Android®. Розроблений адаптивний алгоритм в ручному та автоматичному режимах, дозволяє: - визначати поточний статус апаратного забезпечення (мобільної платформи); - враховувати особливості оброблюваних даних (типи зображень стеганоконтейнера і стеганоконтента); - коригувати параметри роботи основних програмних модулів дослідного стеганоалгорітма (блоку первинної обробки вхідних даних (зображень) і блоку спеціальних перетворень - стеганомодулю). Крім того, проаналізовано і інші параметри обробки зображень, що мають безпосередній вплив на обчислювальну складність всього алгоритму та якість візуалізації зображень контейнерів і стеганоконтента. Розглянута версія алгоритму є дослідною та служить засобом забезпечення безпеки персональних даних (в даному випадку графічної інформації) користувачів, перш за все, мобільних гаджетів. Основні властивості синтезованого алгоритму дозволяють класифікувати його, як програмний засіб забезпечення стеганографічної захисту, що локалізоване для умов внутрішньокадрової обробки зображень. Представлена версія алгоритму вимагає його подальшого вдосконалення і має своєю головною метою підтвердити правильність обраних методів обробки даних та стратегії створення користувальницького інтерфейсу відповідного мобільного додатку.

Ключові слова: обфускація; кодування з перетворенням; зональне кодування; стеганографія.

UDC 336.741.242.1

О НЕКОТОРЫХ ОСОБЕННОСТЯХ КРИПТОГРАФИЧЕСКИХ ВАЛЮТ И ИХ РОЛИ В СОВРЕМЕННЫХ ФИНАНСОВЫХ СИСТЕМАХ

Вячеслав Волошин

ГВУЗ «Приазовский государственный технический университет», ул. Университетская, 7,
г. Мариуполь, 87555, Украина
p@pstu.edu

Рецензент: Александр Холькин, д. физ.-мат. наук, ГВУЗ «Приазовский государственный технический университет», ул. Университетская, 7, г. Мариуполь, 87555, Украина
a.kholkin@gmail.com

Поступила в марте 2018

Аннотация. *В работе представлена аргументация относительно места криптовалют в современной мировой финансовой системе. На основании прямых и косвенных данных показано, что криптовалюта всех видов содержит в себе определенные риски для финансовых и экономических систем, в частности, связанные с кибермошенничеством. Сделан вывод о роли и месте криптовалют, как механизме искусственной стабилизации мирового валютного рынка посредством осознанного манипулирования потребностями людей в условиях, когда глобальное информационное пространство является инструментом для достижения подобных целей, что относится к нарушению прав человека.*

Ключевые слова: *криптовалюта, биткоин, киберпреступления, риски, мировая финансовая система, мировая торговля, блокчейн-операции, финансовые пирамиды в современном киберпространстве.*

1 Введение

В современном финансовом мире криптовалюта позиционируется как объективная данностью, обладающая своими ценными и не очень качествами. В любом случае, мировая общественность признала за криптовалютами право на существование [1-4]. И не беда, что эти технологии проявили себя, в первую очередь, на финансовом поле, в попытках сформировать новое отношение к деньгам [5].

Тем не менее, мировая финансовая система весьма неоднозначно реагирует на появление криптовалюты. Игнорировать ее уже не удается. Но признавать за ней права полноценных денег почти никто не берется. Даже такие страны, как Германия, Япония, Швейцария и США, которые признают за криптовалютой весьма условный аналог денег и пускают их в оборот на своих торговых рынках, очень осторожны в собственной юридической обоснованности этих действий, объясняя их, прежде всего, необходимостью вводить это явление в некоторое регулируемое правовое поле, тем самым снижая риски кибермошенничества [6,7].

Комиссия по финансовым преступлениям США относит биткоин к "децентрализованной валюте" с признаками виртуальности и требует лицензирования подобной деятельности [8, 9]. Китай и некоторые другие страны, по той же причине, ограничили свободное хождение криптовалют, заявив, что биткоин и ему подобные, не являются реальной валютой, и даже наказывают банки, которые обеспечивают ее участие в коммерческих сделках, ссылаясь на их потенциальную криминальную составляющую и способности к киберпреступлениям [10-13]. Какова же реальная цель развития рынка криптовалют?

Анализ результатов последних исследований и публикаций показал неоднозначность условий, в которых существуют все криптографические валюты, в качестве программного продукта, которые обусловлены, теми рисками, которые они привносят в современное общество, а также неопределенностью их правового статуса. В связи с этим ставится задача определения их места и, в равной степени, места функциональности блокчейн операций, как принципиально нового инструментария и связанных с этим рисков.

Целью публикации: - представить место и роль современных криптовалют, как

альтернативного механизма воздействия на мировые финансовые системы, основанного, в том числе, на элементах кибернетических рисков, а также, показать роль блокчейн-операций, как современного высокоэффективного социально-экономического инструмента развития мирового торгового рынка.

2 Основная часть

На сегодня ни одна страна в мире не обладает убедительным законодательством, позволяющим регулировать отношения хотя бы с одним видом криптовалюты, например, в налоговой сфере, или в области объективных финансовых обменных операций, или в области формирования системы денежных обязательств по отношении к криптовалюте (*включая киберпреступления*). Не следует забывать, что за криптовалютой остается *ничем не ограниченное право риска* быть средством для отмывания денег, обладать возможностью уклонения от налогов, влиять на права потребителей товаров, быть способными к уходу от декларирования имеющихся активов всех видов, а также обладать широкой шкалой ничем не обоснованных рисков, связанных с операциями с криптовалютой – купля, продажа, обмен, конвертация [15]. То есть быть предметом киберпреступлений. Тем не менее, тенденции к развитию криптовалюты очевидны.

Пример деятельности ФРС США, в этой сфере деятельности, говорит сам за себя. Многое может сказать то, что эта организация стоит на страже интересов доллара, как мировой валюты, но пока никак не проявляет обеспокоенности по поводу активности биткоина и других криптовалют. Почему? Но есть еще более убедительные примеры. В мире существует 15 компаний, совокупные активы которых превышают размер $ 1 трлн. При этом 11 из них представляют финансовый бизнес США, 3 – Европейский Союз и одна из Японии. Как правило, такие компании в первую очередь должны реагировать на подобные изменения на финансовых рынках, потому, что они могут принести для них колоссальные потери. Но пока от них таких реакций на финансовых рынках мы не видно. И это при том, что криптовалюта не имеет за собой обеспеченности каким-либо товаром или произведенными услугами, то есть является необеспеченной виртуальной валютой. Для уточнения роли необеспеченности денег в мировой экономике можно привести следующие цифры: мировой финансовый долг в 2017 году достиг $ 133 трлн, что в 19-26 раз больше всей массы обеспеченных товарами и услугами денег. Причем, доля США и ее компаний в мировой долговой корзине составляет более $ 17 трлн, доля стран ЕС – $ 15,5 трлн., доля Японии примерно $ 12 трлн. [16].

Прочность современной экономики можно оценивать, например, количеством экономических кризисов, потерями каждого государства и отдельных людей. А можно измерять резистентностью к современным вызовам, одним из которых уже сегодня можно считать появление виртуальных денег в виде криптографических валют, их роли в современных денежных отношениях.

Почему сегодня количество пользователей криптовалюты измеряется миллионами и при этом постоянно растет? Почему периодически появляется информация о том, что биткоин становится субъектом торговых операций то в одной, то в другой компании? Почему дозировано поступает информация о том, что разрешающие организации в отдельных странах дают зеленый свет биткоину, но при этом утверждают, что он, как элемент кибертехнологий должен иметь статус товара, но не денег? Рынок криптовалюты постоянно кем-то периодически «подогревается», причем малыми дозами. Только за 2017 год биткоин вырастал в цене в 20 раз! При этом финансовые биржи не спешат допускать криптовалюту для совершения полноценных операций. Опасаются обвала? Возможно. Возможно потому, что биткоин не настолько силен в межвалютных операциях, как доллар, йена, евро. Но многотысячные процентные прибыли впечатляют? Ни доллар, ни йена, ни евро такой прибыли ни на одной бирже никогда не давали. И давать не будут. Потому, что запредельные ставки реальной валюты могут привести к обвалу торговли по всем другим биржам, торгующим товаром. Деньги так себя вести не могут. Это уже не деньги, а нечто иное. Поэтому на вопрос, криптовалюта это

деньги или нет, можно с уверенностью в 99 % утверждать – нет. И дело здесь вовсе не в вопросах кибербезопасности.

Возможна альтернатива: либо криптовалюта сегодня представляется суперликвидным товаром со специфическим источником происхождения, либо современная экономика сегодня стоит на пороге невиданных изменений (это тот самый 1 %). Прежде всего, изменений относящихся к товарообменным операциям. Из которых скорее всего уйдет свойство эквивалентности по отношению к деньгам и товару.

Трудно сразу понять такие перспективы, но совершенно новые особенности торгового обмена потенциально могут стать явью. Например, когда избыточная масса существующей в мире традиционной валюты (*прежде всего доллара*) будет покрываться с избытком совершенно иной, дорогой, но высоколиквидной валютой (*например, криптовалютой*), которая, поглотив долларовую массу, тем не менее, не станет ее альтернативой по причине своей виртуальности и отношению к производимому товару, но откроет дорогу к обмену на саму себя некоторых промежуточных ликвидных активов.

Но не подобной ли субвалютой, пока товарной, является биткоин? Правда это уже будет не торговля в традиционном смысле. Возможно в ее основе, на первых порах, будет «перегретый», неравномерно распределенными товарами современный мировой рынок, который со временем уменьшится в объеме, по мере потребления всех этих товаров.

Собственно технологии блокчейна, основа появления криптовалюты, безусловно, найдут свое применение в самых различных областях, хотя бы потому, что в их основе лежит попытка при помощи информационных технологий создать альтернативную современным банкам систему доверительных отношений без посредников.

Именно эта способность блокчейн-операций делает их интересными в самых различных областях деятельности. Например, в области Интернет-вещей, о чем пишет в книге «Революция блокчейна» Алекс Тэпскотт. В компании Intel создана киберплатформа, при помощи которой можно отслеживать сети поставок морепродуктов. Группа компаний AusPost, PwC, Alibaba Group, используя методику блокчейна, создала систему «Food Trust Framework» для управления цепями поставок пищевых продуктов с одновременным объективным контролем качества и прозрачности поставок. Корпорация Bosch использует технологии блокчейна для предупреждения кибермошенничества при контроле показаний счетчика автопробега автомобилей. Широкие возможности для применения этих технологий открываются в логистике и маркетинговых операциях. Уже из этого краткого перечня можно судить об универсальности подобных технологий, как способа обмена ценностями и материальными потоками, благодаря возможностям блокчейна в сферах создания новых типов рыночных отношений и рынков товаров с функциями рынка ресурсной оптимизации.

Но это будет уже совершенно другая экономика, являющаяся продуктом глобальной информационных технологий. Это может быть экономика репутаций, взаимного внимания и доверия, своеобразная «услужливая» экономика, адаптированная под потребителя. Экономика, которая, потенциально, не будет нуждаться в государстве (*регуляторе*). Основанная на совершенно иных, пока не совсем понятных, правилах и условиях. Такая киберэкономика сможет иметь право формироваться, как переходная альтернатива от современной экономики, основанной на бесконтрольном печатании денег и неуправляемом развитии необеспеченности мировой валюты товарами и услугами. В ее основу может быть положено общее правило: каждая торговая операция должна понемногу «поглощать» часть массива прошлых необеспеченных денег. При этом, в качестве такого «поглотителя» возможно и будет криптовалюта. Такая экономика может являться промежуточным звеном на пути перехода к иной децентрализованной мировой экономике, ориентированной на правила существования глобального информационного пространства, со всеми его достоинствами и недостатками.

И, безусловно, велика роль денег в этих изменениях. Одним из критериев здесь может быть готовность каждой страны (см. табл. 1), входящей в систему мировой торговли к отказу от торговли за наличность на самом низком уровне – на уровне обывателя в пользу их цифровых аналогов [16]. Готовность страны к отказу от наличности в финансовых расчетах в пользу цифровых аналогов денег показывает степень ее адаптации к новым правилам тор-

говли и, в свою очередь, определяется несколькими критериями. Прежде всего, это индекс цифровой эволюции в экономике страны. Он характеризует возможности финансовых организаций управлять деньгами, представленными в цифровых кодах при условии сохранения и поддержания их реальной стоимости. То есть, способности к валютному обмену при помощи компьютерных программ и цифровых кодов. Но при этом важна готовность населения к переходу к цифровым деньгам, наличию электронных кабинетов, кошельков, формирование соответствующей инфраструктуры (*например, для интернет торговли*) и т.д. Абсолютная "цена" наличности, это индекс паритетности между наличными деньгами и их цифровыми аналогами.

Таблица 1 – Готовность стран к отказу от наличных денег

№	Государство	Абсолютная "цена" наличности	Индекс цифровой эволюции	Характеристика перехода к цифровым деньгам
1	Швеция	0,4	57	Сохраняется возможность для создания реальной стоимости денег за счет ускорения перехода к цифровым деньгам
2	Дания	0,5	51	
3	Кения	0,3	18	
4	Турция	0,65	33	
5	Южная Корея	4,4	51	Устойчивость экономики к переходу на цифровые деньги при сохранении их реальной стоимости
6	Япония	5,5	47	
7	США	7,5	54	
8	Филиппины	5,8	20	
9	Египет	8,5	18	
10	Россия	8,5	26	
11	Бельгия	15	45	Максимальный потенциал для создания реальной стоимости за счет приоритета инвестиций для перехода в цифровую плоскость
12	Германия	18	48	
13	Франция	36	48	
14	Мексика	26	28	
15	Индия	82	22	

Риски кибернетического характера, связанные с ликвидацией кэш-массы в государстве в пользу их цифровых аналогов, зависят от скорости этого процесса (ΔК), отношению к всей денежной массе (M_k) и продолжительности во времени (T). Примем допущение о рисках, связанных с соответствующей финансовой неустойчивостью экономики в виде условной сборки Уитни. Для области финансовой неустойчивости в виде сборки Уитни, связанной с уменьшением кэша, предложена параметрическая зависимость вида $\Delta K = 1{,}3 \cdot T^2 - 5{,}6 \cdot T + 5{,}9$ (рис. 1, г). Тогда каждая из экономик будет по своему соотноситься со временем и скоростью ликвидации кэша. Всего можно выделить три группы стран по их адаптации к цифровым денежным технологиям в плане удаления кэша.

К первой группе следует отнести государства, которым не нужны дополнительные усилия к переходу на электронный финансовый оборот (Китай, Дания, Финляндия, Новая Зеландия (см. рис. 1, а)). Эти страны проявляют осторожность в электронном обороте, ориентируясь на партнеров по международной торговле, таким образом, чтобы не дестабилизировать своими инициативами собственную «твердую» валюту. При этом некоторые другие страны из этой же группы (Южная Африка, Турция и др.) пока не готовы к таком переходу, но сохраняют возможность при условии внешней поддержки их национальных валют (табл. 1). Однако, очевидно, что не для всех такая поддержка возможна.

Вторая группа стран весьма адаптирована к возможностям перехода к электронным деньгам (Великобритания, США, Япония, Голландия, Южная Корея и др). Высокий индекс цифровой эволюции в финансовой сфере при устойчивости цены собственной валюты тому подтверждение (рис. 1б и Табл. 1).

Третья группа стран, прежде всего лидеры Европейского Союза, готовы к такому перехо-

ду, однако ограничены возможностями некоторых своих партнеров по Союзу (Испанией, Чехией, Польшей, Литвой и др.), которым нужны существенные инвестиции для поддержания стоимости их национальных валют. К этой же группе можно отнести и Украину с ее неустойчивой гривной (рис. 1, в).

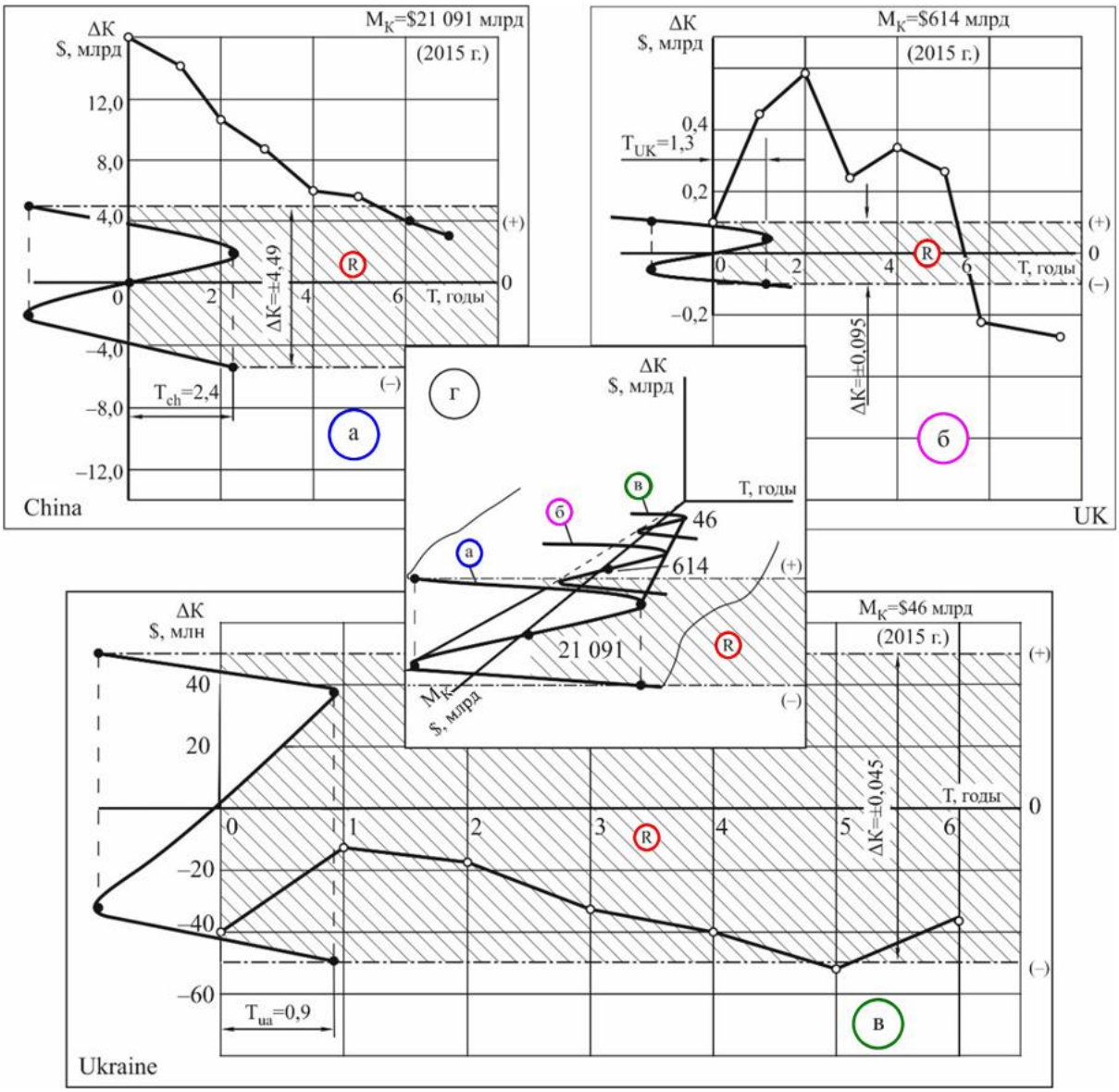


Рис. 1 – Риски, связанные с уменьшением кэш-массы для разных государств:
а) – избавляющихся от кэша; б) – балансирующих в зоне финансовой устойчивости; в) – балансирующих в зоне финансовой неустойчивости; г) – поверхность неустойчивости при управлении финансами; R – области финансовых рисков.

В целом следует предполагать, что неизбежность отказа от наличных денег в пользу их цифровых аналогов диктуется всеми изменениями, которые претерпевает современная торговля, ее растущие объемы и необходимые для этого скорости финансовых оборотов. Следует ожидать, что это может привести к существенным изменениям в самих торговых процедурах и изменить сущность балансов в торговле, что в свою очередь, может привести к непредсказуемым последствиям, не всегда лояльным для многих стран, особенно с «недоразвитой», в настоящем понимании, экономикой.

Вернемся к криптовалютам и их роли в современной экономике, связанной с возможностями мировой торговли. Имеет право на существование следующая гипотеза относительно ее перспектив (рис. 2).

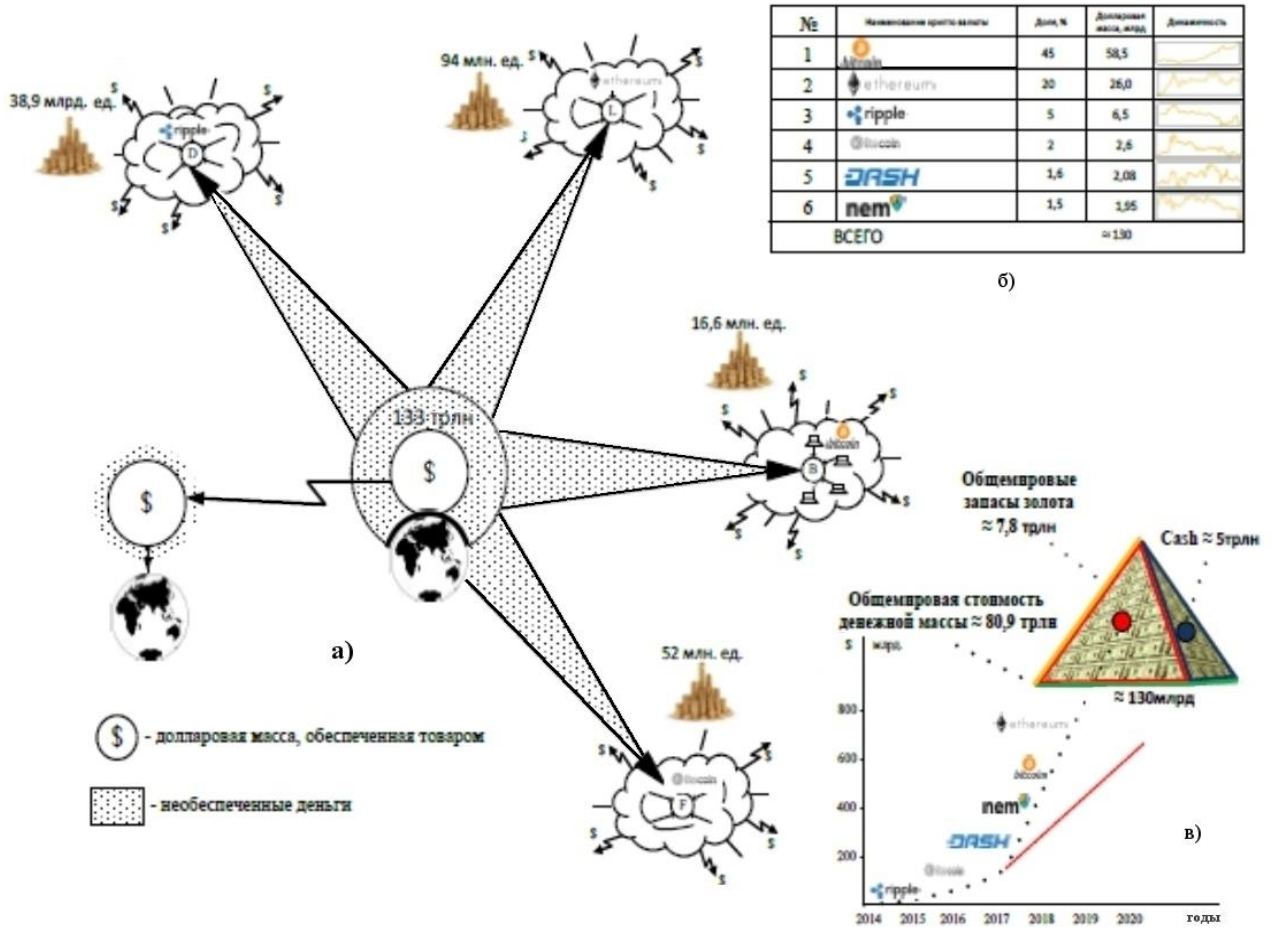


Рис. 2 – Место и роль криптовалюты в современной мировой экономике (*гипотеза*):
а) – модель развития; б) – весовые функции известных криптовалют и их долларовое обеспечение; в) – динамика роста долларового обеспечения криптовалют (прогноз).

Очень вероятно то, что биткоин, это развивающийся проект, авторство которого в конечном результате принадлежит тем, кто порождает лавинообразные процессы появления необеспеченных денег. Цель – уменьшить общее давление на мировую экономику со стороны совокупной массы необеспеченных денег, и в первую очередь, доллара. В этом случае можно предположить, что биткоин как губка напитавшаяся массой долларов, вращающихся в мировой финансовой системе, в некоторый момент "лопнет", и подомнет под себя большое количество необеспеченной денежной массы, тем самым, подняв условную ценность оставшейся ее части до необходимого в мировой торговле уровня. Это шанс остаться доллару "на плаву" и укрепить эту валюту еще на долгое время. Это шанс не «развалить» мировую торговлю и избежать потрясений, подобных прошлым мировым войнам. И если сегодня совокупная криптовалюта занимает нишу ориентировочно в $ 420 млрд, то через два-три года этот уровень может составить более $ 1 трлн., а по некоторым оценкам $ 3-4 трлн [7], что уже сопоставимо с суммами необеспеченных денег (см. табл. 2) [16]. Возможно этим и объясняется такая реакция на криптовалюты со стороны ФРС и крупнейших компаний мира? Опыт создания таких пирамид уже давно накоплен, хотя и дискредитирован во всем мире. Так что грех им "не воспользоваться" на благо "золотого тельца". Результатом может быть появление нового «крепкого» доллара, который по прежнему будет составлять основу мировой торговли. Для этого необходимо соблюдать определенные условия, обеспечивающих поддержание проекта криптовалют, а именно:

1 - нельзя объявлять биткоин общепризнанной валютой, по крайней мере, официально, но следует поддерживать к ней подобный интерес, как к неофициальной валюте (*в настоящее время условие соблюдается*);

2 - способствовать вовлечению в проект максимального количества людей, обладающих свободными активами и валютой (*в настоящее время условие соблюдается*);

3 - стимулировать возможности для накопления криптовалют у резидентов, с обеспечением условия необратимости (*соблюдается периодически*);

4 - по возможности, ограничивать развитие свободной торговли за биткоин, но одновременно и не запрещать, тем самым способствуя их накоплению у резидентов (*пока соблюдается*);

5 - стремиться не подорвать авторитет доллара в этой игре (*условие соблюдается на правах роста ликвидности биткоина*);

- уходить от официального сопоставления доллара и биткоина.

Таблица 2 – Весовые функции популярных криптовалют на конец 2017 года

№	Наименование криптовалюты	Курс, [$/монета]	Эмиссионное ограничение, [число монет]	Потенциальный размер кэша, [$]
1	Bitcoin (BTC)	19000	21 млн	400 млрд
2	Darkcoin (DASH)	14	22 млн	300 млн
3	Litecoin (LTC)	33	84 млн	2,77 млрд
4	Feathercoin (FTC)	0,43	185 млн	80 млн
5	Primecoin (XPM)	3,56	14 млн	50 млн
6	Peercoin (PPC)	6,04	48 млн	290 млн
7	Namecoin (NMC)	8,0	21 млн	168 млн
8	Freicoin (FRC)	0,7	100 млн	70 млн
9	Eathereum (ETH)	460	28 млн	13 млрд
	ВСЕГО			**≈ 416 млрд**

Таким образом, почти все эти условия для реализации такого проекта до сих пор соблюдаются. Намеренно или спонтанно – пока остается не понятным. Но, как гипотеза самопроизвольного или запланированного процесса по указанному алгоритму, она имеет право на существование.

3 Выводы

В работе обобщены результаты анализа различных источников в области криптовалют и кибермошенничества и определены место и роль современных криптовалют, как альтернативного механизма воздействия на мировые финансовые системы.

Обращено внимание на факт существования мнения относительно криптовалют, как о глобальных финансовых пирамидах, основанных на непропорциональности доходов и расходов, которые стоят за денежными оборотами. Обозначена и другая точка зрения относительно криптовалют, как о некоторых глобальных финансовых достижениях, основанных не на пропорциональности доходов и расходов, а на эксплуатации негативных качеств человека, как субъекта новых видов кибермошенничества в рамках глобального информационного пространства. Существование подобных мнений может иметь место, в частности, если современное общество будет подведено к пониманию того, что криптовалюты имеют перспективу движения в качестве новой валюты с совершенно новыми свойствами и качествами. Либо в виде иных проектов, на которые сегодня так щедра мировая экономика.

Ссылки

1. Tapskott, A. Tekhnologiya blokchein. To, chto dvizhet finansovoi revolyutsiei segodnya / A. Tapskott, D. Tapskott. – M.: EKSMO, 2017. – 448 p.
2. Trepper, A. The People's Money Bitcoin / Adam Trepper. – [s. l.], 2015. – 59 p.
3. Distributed ledger technology: Blackett review / ed. Mark Peplow; Crown. – London, 2016. – 88 p.
4. Akst, R. Sem' sekretov bitkoina ili bitkoin za chas / R. Akst. – Ekaterinburg: Izdatel'skie resheniya, 2017. – 50 p.

5. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash [Electronic resource] / Satoshi Nakamoto. – [s. l], 2008 – 9 p. – Mode of access: http://nakamotoinstitute.org/static/docs/bitcoin.pdf
6. Shlygin, I. Bitkoin – glavnaya kriptovalyuta, chto za nim stoit [Elektronnyi resurs] / Ivan Shlygin. – Rezhim dostupa: https://fomag.ru/news/bitkoin-glavnaya-kriptovalyuta-chto-za-nim-stoit
7. Nigmatullin, T. F'yuchers na bitkoin lopnet puzyr' na rynke kriptovalyut [Elektronnyi resurs] / Timur Nigmatullin. – Rezhim dostupa: https://fomag.ru/news/fyuchers-na-bitkoin-lopnet-puzyr-na-rynke-kriptovalyut/
8. Novosti. SShA. Finansy, banki [Elektronnyi resurs] / Polpred. – Rezhim dostupa: http://www.usa.polpred.ru/?cnt=151%3Fcnt%3D151%3Fcnt%3D151%3Fcnt%3D151%3Fcnt%3D250&ns=1&sector=11&page=2.
9. BitSoin (Bitkoin) – osobennosti, pokupka/prodazha, analitika [Elektronnyi resurs] / Money maker group. – Rezhim dostupa: https://mmgp.ru/showthread.php?t=72688
10. Novosti kriptovalyut [Elektronnyi resurs] / ChAO "Segodnya Mul'timedia". – Rezhim dostupa: https://www.segodnya.ua/economics/kriptovalyuta.html
11. Xarper. Evropol analiziruet Bitcoin prestupleniya v novom doklade [Elektronnyi resurs] / Xarper; Bits Media. – Rezhim dostupa: https://bits.media/news/evropol-analiziruet-bitcoin-prestupleniya-v-novom-doklade/
12. Top-9 krupnykh makhinatsii s bitkoinami [Elektronnyi resurs] / Happy Coin Club. – Rezhim dostupa: https://happycoin.club/top-9-krupnyih-mahinatsiy-s-bitkoinami.
13. Vlasti Kitaya reshili zapretit' proizvodstvo bitkoinov. Na stranu prikhoditsya 80 % mainingovykh moshchnostei mira [Elektronnyi resurs] / Meduza. – Rezhim dostupa: https://meduza.io/feature/2018/01/10/vlasti-kitaya-reshili-zapretit-proizvodstvo-bitkoinov-na-stranu-prihoditsya-80-mayningovyh-moschnostey-mira
14. Kak zarabotat' na Bitcoin? [Elektronnyi resurs] / Forex Club Libertex. – Rezhim dostupa: https://promo.fxclub.org/lp/ru-ru/how-to-trade-bitcoin
15. Rynok Foreks: novosti, prognozy i analitika [Elektronnyi resurs] / ForexStandard. – Rezhim dostupa: https://forexstandard.ru

Reviewer: Aleksandr Kholkin, Prof., Doctor of physico-mathematical sciences, Pryazovskyi State Technical University, Mariupol, Ukraine. E-mail: a.kholkin@gmail.com

Received: March 2018.

Authors:
Vyacheslav Voloshin, Doctor of Sciences (Engineering), Prof., Rector of the Pryazovskyi State Technical University, Mariupol, Ukraine. E-mail: p@pstu.edu

On specifics of cryptographic currencies and their role in modern financial systems.

Abstract. The paper presents argumentation on the place of crypto currency in the modern world financial system. Based on direct and indirect data, it is shown that all types of crypto currency contain certain risks for financial and economic systems, in particular those related to cyber fraud. The conclusion is made about the role and place of the crypto currency as a tool for the artificial stabilization of the world currency market through conscious manipulation of people's needs in conditions when the global information space serves as an instrument for achieving the above goals, which refers to the violation of human rights.

Keywords: crypto currency, bitcoin, cybercrime, risks, the world financial system, block-chain operations, financial pyramids in modern cyberspace.

Рецензент: Олександр Холькін, доктор фізико-математичних наук, проф., ДВНЗ «Приазовський державний технічний університет», Маріуполь, Україна. E-mail: a.kholkin@gmail.com

Надійшло: Березень 2018.

Автори:
В'ячеслав Волошин, д.т.н., проф., ректор ДВНЗ «Приазовський державний технічний університет», Маріуполь, Україна.
E-mail: p@pstu.edu

Про деякі особливості криптографічних валют та їх ролі в сучасних фінансових системах.

Анотація. В роботі представлена аргументація щодо місця криптовалют у сучасній світовій фінансовій системі. На підставі прямих і непрямих даних показано, що криптовалюта всіх видів містить в собі певні ризики для фінансових і економічних систем, зокрема, пов'язані з кібершахрайством. Зроблено висновок про роль і місце криптовалют, як механізму штучної стабілізації світового валютного ринку за допомогою усвідомленого маніпулювання потребами людей в умовах, коли глобальний інформаційний простір служить інструментом для досягнення подібних цілей, що відноситься до порушення прав людини.

Ключові слова: криптовалюта, біткоін, кіберзлочини, ризики, світова фінансова система, блокчейн-операції, фінансові піраміди в сучасному кіберпросторі.

UDC 004.056.55

HIDING DATA IN THE FILE STRUCTURE

A. Kuznetsov, K. Shekhanin, A. Kolgatin, K. Kuznetsova, Ye. Demenko

V. N. Karazin Kharkiv National University, 4 Svobody Sq., Kharkiv, 61022, Ukraine
kuznetsov@karazin.ua, kyryl.shekhanin@nure.ua, kolgatin-a@yandex.ua, kate.kuznetsova.2000@gmail.com, demenjay@gmail.com

Reviewer: Serhii Toliupa, Doctor of Sciences (Engineering), Full Prof., Taras Shevchenko National University of Kiev, 81 Lomonosova St., Kyiv, 03189, Ukraine.
tolupa@i.ua

Received on March 2018

Abstract. *In this paper, the methods of steganography hiding of information in a file system structure is investigated. Namely, the structure of the FAT file system (File Allocation Table) and methods of hiding information messages, which are based on repositioning separate clusters of cover files. A new method is proposed that, unlike the known ones, changes the order of alternation of clusters in each cover file, which allows to further hide a certain informational message, that is, to increase the capacity of the hidden channel. It was confirmed that the results of the data concealment and deletion procedures largely depend on the number of clusters with which it is necessary to carry out the appropriate transformations. It is noted that the extraction procedure is performed much faster than hiding the message. The proposed method is implemented programmatically, the results of experimental researches confirmed the adequacy of the theoretical conclusions and recommendations.*

Keywords*: steganography; hiding information data; file system.*

1 Introduction

Steganographic methods of information protection become, in recent years, increasingly popular and widespread [1-2]. In particular, this is due to the emergence of the latest technologies of hidden communication messages in artificially created containers, redundancy in which is generated by technical features of storage, processing and/or transmission of digital data [3-14]. Namely, methods of network steganography as a carrier (container) use transmitted over the network packet or a set of data packets [3-6]. In the 3D steganography, informational messages hide into artificial excess of digital 3D object models, for example, in the retina of surfaces, holograms, etc. [7-9]. The construction of hidden cluster channels is based on the use of data storage features in modern file systems [10-14]. The last direction is researched in this paper, in detail, researched of methods of steganography hiding of information in the file system structure.

2 Modern file systems

The file system is the procedure established, which determines the way of organizing, storing and naming data on the storage media in computer systems, as well as in other electronic equipment: digital cameras, mobile phones, etc. [15-17]. The file system determines the format of the content and the method of physical storage of information, which is grouped into files. The specific file system defines the size of file names and (directories), the maximum possible file size, and defines the set of file attributes. Some file systems provide service capabilities such as access control or file encryption.

The main functions of the file system are aimed at solving the following tasks: naming files; application file interface; displaying the logical model of the file system on the physical organization of the data warehouse; organization of file system stability to power failure, hardware and software errors; content of the file parameters necessary for its proper interaction with other system objects (kernel, application, etc.).

In multi-user systems, there is another task: protection of one user's files from unauthorized access of another user, as well as collaborative work with files, for example, when a file is opened by one user, for others, the same file will temporarily be available in read-only mode.

The greatest development in computing technology has traditionally been disk drives, the structure of which is generally presented in Fig. 2. Data on disk drives are recorded on tracks. The set of tracks is divided into geometric sectors, while part of the path of a specific geometric sector is called the track sector. The main logical unit of data storage in the file allocation table for disk file systems is a cluster.

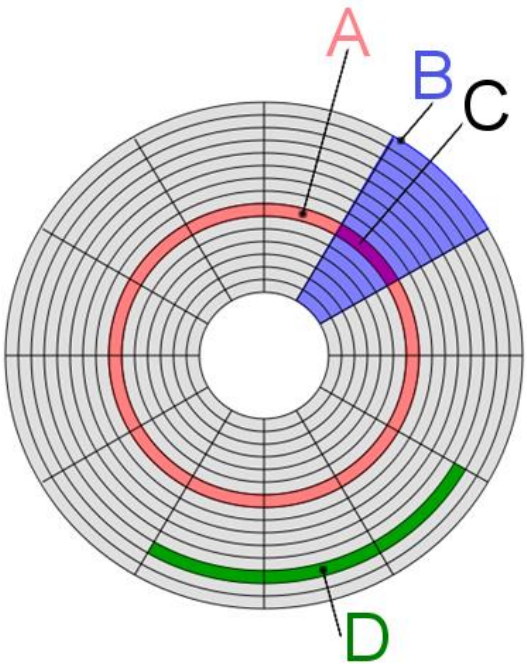


Fig. 1 – Structure of disc:
(A) - track; (B) - geometric sector;
(C) - sector of the track; (D) - cluster.

The cluster is a logical storage unit in a file allocation table that combines a group of sectors of a track. For example, on the 256-byte-sized sector, a 256-byte cluster contains one sector, while the 2-kilobyte cluster contains eight sectors. As a rule, a cluster is the smallest disk space that can be allocated to store a file.

The concept cluster is used in file systems, FAT, NTFS, and others. Other file systems use similar concepts (zones in the Minix, blocks in the Unix, etc.). On some Linux file systems (ReiserFS, Reiser4, Btrfs), BSD (FreeBSD UFS2), the last block of a file can be divided into subfragments, which can be placed "tails" of other files. In NTFS, small files can be written to the Master File Table (*MFT*). The small cluster is best suited for small files. So, this way is more economical. A large cluster allows you to achieve higher speeds, but in small files, the place will be used irrationally (*many sectors will not be fully filled up but will be considered busy*).

Disk file systems are usually stream-oriented. Files in stream-oriented file systems are a sequence of bits, often providing functions such as read, write, change data and control access. The most common current stream-oriented file systems are FAT (*File Allocation Table*) - its three different types (*FAT16\32\64*) and NTFS (*New Technology File System*). Given the complete openness of the file system specification in this paper, only FAT is considered below.

The structure of the FAT consists of five parts: Volume ID; FAT - tables (*two examples*); Clusters (*data files*); Root directory.

Volume ID located at the beginning of the disk partition of the FAT file system. It is required for the initial boot of the device. It also contains information about the parameters of the file system.

File Allocation Table is intended to indicate clusters of individual files. The disk data area is separated into clusters - blocks that are sized when formatting a disk. Each file and directory occupy one or more clusters. Thus, clusters of chains are formed. In the file allocation table, each cluster is marked in a special way. The pointer size in bits for each cluster is specified in the file system name. For example, for the FAT32 file system, the size of the pointer is 32 bites. There are three types of cluster pointers:

- free cluster is a cluster in which new files and directories will be recorded;.
- busy cluster - the pointer indicates the next cluster in the chain. If the chain of clusters is over, then the cluster is marked with a special value (0xFFFFFF0F in hex);
- Bad block - cluster with access errors. Indicated when formatting the drive to disable later access to it.

Damage to the file allocation table completely destroys the file system structure, so two copies of the table are always stored on the disk.

Clusters (data files) - the data area that is placed directly after the last FAT table. The FAT directory (folder, directory) is an ordinary file marked with a special attribute. The data of such a file in any version of FAT is a chain of 32-byte file records (*directory entries*). The catalog cannot contain two files with the same names. If the disk validation program detects an artificially created pair of files with the same name in one directory, one of them is renamed.

Root directory – the disk area in which the root directory information is located. Its size is limited, so in the root directory of the disk can be no more than 512 files and subdirectories.

The main advantage of the FAT file system is its simplicity and compatibility with outdated operating systems. For this file system, there is a many detailed open documentations. A breach in the system often lead to damage to one or more files. However, in case of serious damage, it is much easier to restore information than NTFS.

3 Steganographic methods of hiding information in the file system structure

The simplest steganographic methods of hiding information in the structure of the file system are discussed in [10,11]. They use free clusters (or certain service data fields) to record a hidden message, but this method is unreliable [13,14]. Other methods, such as [12-14], are based on the use of multiple cover files and hiding an informational message by changing the relative positions of the clusters of different cover files one to another.

The hidden data is presented in the form of a bit array:

$$M = \{b_0, b_1, ..., b_{n-1}\},\ b_i \in \{0,1\}.$$

On the device $p = 2^m$, $m \in N$ cover files are selected:

$$F_0, F_1, ..., F_{p-1}.$$

The order of the clustering of the cover files will hide the information message, that is, after the embedding, the cover files cannot be deleted, moved or modified. The natural number m and names of the cover files is the secret key. Also important is the order of cover files [14].

An array of cluster numbers for cover files is formed:

$$C = \begin{pmatrix} c_{0,0} & c_{0,1} & \dots & c_{0,L_0-1} & & & \\ c_{1,0} & c_{1,1} & \dots & \dots & \dots & \dots & c_{1,L_1-1} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ c_{p-1,0} & c_{p-1,1} & \dots & \dots & c_{p-1,L_{p-1}-1} & & \end{pmatrix}, \quad (1)$$

where each row of the array contains the cluster numbers of the corresponding file. For example, the file F_i corresponds to the i line of the array C, that is, the cluster numbers of the i covering file can be represented as an array

$$C_{F_i} = \{c_{i,0}, c_{i,1}, ..., c_{i,L_i-1}\}$$

where L_i - clusters number in i cover file.

If, when hiding the information, it is necessary to save without changing the contents of the cover files, then it is necessary that the conditions are fulfilled: $\forall i : L_i \geq k$, $k = n/m$. An array D of empty file system cluster numbers is formed: $D = \{c_1, c_2, ..., c_{L_D}\}$, And $c_1 < c_2 < ... < c_{L_D}$.

The number L_D is the number of empty clusters of the file system, and it is required that the condition is fulfilled:

$$L_D \geq \sum_{i=0}^{p-1} L_i.$$

Information message M is separated into blocks by m bits each: $M = \{B_1, B_2, ..., B_k\}$, that $k = \lceil n/m \rceil$ and if $k = n/m$, then $B_1 = \{b_0, b_1, ..., b_{m-1}\}$, $B_2 = \{b_m, b_{m+1}, ..., b_{2m-1}\}$, …,

$B_k = \{b_{(k-1)m}, b_{(k-1)(m+1)}, ..., b_{km-1}\}$. If $k < n/m$, then the last block is supplemented by zeros $B_1 = \{b_0, b_1, ..., b_{m-1}\}$, $B_2 = \{b_m, b_{m+1}, ..., b_{2m-1}\}$, …, $B_k = \{b_{(k-1)m}, b_{(k-1)(m+1)}, ..., b_{n-1}, \underbrace{0, 0, ..., 0}_{km-n}\}$.

Each block B_i, $i = 1, 2, ..., k$ is interpreted to the natural number $\forall i : 0 \le B_i \le p-1$. Each of the natural number. B_i., $i = 1, 2, ..., k$ is interpreted to the number of cover files from files array $F_0, F_1, ..., F_{p-1}$.

All clusters of the cover files are overwritten to empty clusters, that is, the D array is filled with cluster numbers from the array C. The order of overwriting clusters of cover files corresponds to a sequence of natural numbers $\{B_1, B_2, ..., B_k\}$, which are conditioned by the information message. For example, the first empty cluster overwrites the first cluster of the cover file with number B_1, in the second empty cluster overwrites the next cluster of the cover file with the number B_2 etc. The natural numbers B_i may coincide, and in this case, the clusters of the same cover file with the number B_i are written. In order to enhance the protection, various techniques can be used, for example [14], the initial value of B_0, is selected, and the order of overwriting the clusters of the covering files is given by a sequence of positive integers $\{N_1, N_2, ..., N_k\}$,

$$N_i = B_{i-1} + B_i \bmod p,\ 0 \le N_i \le p-1.$$

After, the first empty cluster overwrites the first cluster of the cover file with the number N_1, in the second empty cluster - the next cluster of the cover file with the number N_2 etc. At the result of the algorithm, the first k empty clusters of the file system will be written by clusters of the cover files. So the condition $k \le L_D$ must be fulfilled.

To extract the information message M the array D of the clusters of the cover files is formed: $D = \{c_1, c_2, ..., c_{L_D}\}$, and $c_1 < c_2 < ... < c_{L_D}$. Each cluster number in this array is correlated with only one cluster of the cover file. This correspondence is determined by the logic of embedding information and is used to extract data. In this case, a sequence of natural numbers is formed $\{B_1, B_2, ..., B_k\}$, which correspond to the blocks of the informational message:

$$B_1 = \{b_0, b_1, ..., b_{m-1}\},\ B_2 = \{b_m, b_{m+1}, ..., b_{2m-1}\},\ ...,\ B_k = \{b_{(k-1)m}, b_{(k-1)(m+1)}, ..., b_{km-1}\}.$$

The informative message is calculated from these bit blocks

$$M = \{b_0, b_1, ..., b_{n-1}\},\ b_i \in \{0,1\}.$$

If $k < n/m$, then the last block is "cut" - its last $km-n$ bits are not used.

The disadvantage of this method is a small size of hidden data, which depends on the number of cover files and the size of the one cluster at the file system. Each cluster of cover files can contain $\log_2 p = m$ information bits. In this paper, we propose a new steganographic method of hiding information in the structure of the file system, which, in contrast to the one discussed, further modifies the order of the cluster alternation in each cover file. The improved method allows an increase in the amount of hidden information is achieved.

4 Proposed method

The proposed method of steganographic hiding of data in cluster file systems is based on the use of several cover files (*as in the prototype method*) and the hiding of a secret message by changing the relative positions of clusters of different cover files one to the other and, unlike the known methods, the order of alternating clusters in each cover file. The hidden data is represented as the bit array:

$$M = \{b_0^*, b_1^*, ..., b_{L_1+L_2+...+L_{p-1}-1}^*, b_0, b_1, ..., b_{n-1}\},\ b_i^*, b_i \in \{0,1\}.$$

On the device $p = 2^m$, $m \in N$ cover files are selected: $F_0, F_1, ..., F_{p-1}$.

The array of cluster numbers of cover files is formed as (1). For each cover file, the order of alternating clusters in each cover file is changed. The order of alternation is given by the information sequence M. To do this, p bit arrays of information bits are formed

$$M_1 = \{b^*_0, b^*_1, ..., b^*_{L_1-1}\},$$
$$M_2 = \{b^*_{L_1}, b^*_{L_1+1}, ..., b^*_{L_1+L_2-1}\},$$
$$\dots,$$
$$M_{L_p} = \{b^*_{L_1+L_2+...+L_{p-1}}, b^*_{L_1+L_2+...+L_{p-1}+1}, ..., b^*_{L_1+L_2+...+L_{p-1}-1}\},$$

each of which is mapped to an array of cluster numbers of files

$$C_{F_1} = \{c_{1,0}, c_{1,1}, ..., c_{1,L_1-1}\},$$
$$C_{F_2} = \{c_{2,0}, c_{2,1}, ..., c_{2,L_2-1}\},$$
$$\dots,$$
$$C_{F_p} = \{c_{p,0}, c_{p,1}, ..., c_{p,L_p-1}\}.$$

The clusters of each cover file are reordered, that is, the cluster numbers in each of C_{F_1}, C_{F_2}, … , C_{F_p} arrays change their alternation in accordance with the values of the bit arrays M_1, M_2, … , M_{L_p}. As a result, new arrays of cluster numbers are obtained $C^*_{F_1}$, $C^*_{F_2}$, … , $C^*_{F_p}$. Reordering clusters in each cover file can be done in different ways. For example, by splitting down all numbers $\{c_{i,0}, c_{i,1}, ..., c_{i,L_i-1}\}$ into two halves and comparing each half with the value of the information bit $b^*_j = 1$, $L_1 + L_2 + ... + L_{i-1} - 1 < j \le L_1 + L_2 + ... + L_i - 1$, on a j position in array $C^*_{F_i}$ place a cluster from the first half of ordered numbers, if $b^*_0 = 0$ - from the second half.

Formed in this way arrays $C^*_{F_i} = \{c^*_{i,0}, c^*_{i,1}, ..., c^*_{i,L_i-1}\}$ reordered numbers of cover files form an array of

$$C^* = \begin{pmatrix} c^*_{0,0} & c^*_{0,1} & \dots & c^*_{0,L_0-1} & & & \\ c^*_{1,0} & c^*_{1,1} & \dots & \dots & \dots & \dots & c^*_{1,L_1-1} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ c^*_{p-1,0} & c^*_{p-1,1} & \dots & \dots & c^*_{p-1,L_{p-1}-1} & & \end{pmatrix}.$$

Changing the alternation of clusters in each cover file allows you to hide the first $L_1 + L_2 + ... + L_{p-1}$ information bits from the array M, that is, the information sequence $\{b^*_0, b^*_1, ..., b^*_{L_1+L_2+...+L_{p-1}-1}\}$. The rest of the information bits are hidden in the same way as in the above prototype method [14].

The array D of empty file system clusters is formed: $D = \{c_1, c_2, ..., c_{L_D}\}$, $c_1 < c_2 < ... < c_{L_D}$. The sequence of information bits $\{b_0, b_1, ..., b_{n-1}\}$ is separated into blocks by m bits each: $\{B_1, B_2, ..., B_k\}$, Each block B_i, $i = 1, 2, ..., k$ s interpreted as a natural number, i.e. $\forall i : 0 \le B_i \le p-1$. Each natural number B_i, $i = 1, 2, ..., k$ is interpreted as the number of the cover file from the set of files $F_0, F_1, ..., F_{p-1}$. All cluster of the cover files are overwritten in empty clusters, that is, the array D is filled with cluster numbers from the array C^* (reordered clusters, that is, with the alternating cluster interchanges in each cover file). The order of rewrite of cluster covers files corresponds to a sequence of natural numbers $\{B_1, B_2, ..., B_k\}$, that are specified by the information message. For example, the first empty cluster overwrites the first cluster of the cover file with number B_1, in the second empty cluster, the next cluster of the cover file with the number B_2

etc. The natural numbers B_i may coincide, and in this case, the regular clusters of the same cover file with the number B_i are written. As a result, the first k empty clusters of the file system will be recorded by the clusters of the cover files. To extract the information message M, the array D of the cluster numbers of the cover files is formed: $D=\{c_1,c_2,...,c_{L_D}\}$. Each cluster number in this array is correlated with only one cluster of the cover file. In this case, a sequence of natural numbers $\{B_1,B_2,...,B_k\}$, is formed that correspond to the blocks of the informational message, i.e. the information sequence $\{b_0,b_1,...,b_{n-1}\}$, $b_i\in\{0,1\}$ is formed. Then the information sequence is extracted

$$\{b_0^*,b_1^*,...,b_{L_1+L_2+...+L_{p-1}-1}^*\},\ b_i^*\in\{0,1\}.$$

For this purpose, the arrays $C_{F_i}^*=\{c_{i,0}^*,c_{i,1}^*,...,c_{i,L_i-1}^*\}$ of the cluster numbers of each cover file are analyzed. The extraction rule corresponds to the logic of hiding. For example, the splitting of all ordered $\{c_{i,0},c_{i,1},...,c_{i,L_i-1}\}$ numbers into two halves and the matching of each half with the value of the information bit can be applied. For example, if there is a cluster on the j position in array $C_{F_i}^*$ on the first half of an array of ordered numbers $\{c_{i,0},c_{i,1},...,c_{i,L_i-1}\}$, accepts $b_j^*=1$. If the second half accepts $b_0^*=0$. Thus, due to the additional change in the order of alternation of clusters in each cover file, it is possible to increase the size of hidden information. In particular, in comparison with the prototype method, it is possible to hide one bit per cluster of cover files in an additional way.

The proposed method was implemented programmatically, experimental research of its effectiveness was conducted. Fig. 2 and 3 contain results of the comparative analysis of the built-in data capacity by a base method and the method offered in this work are given.

Fig 2 shows the dependence of the size of the steganograms on the size of the cluster of cover files. As it is seen, that is doubling the bandwidth of steganogram.

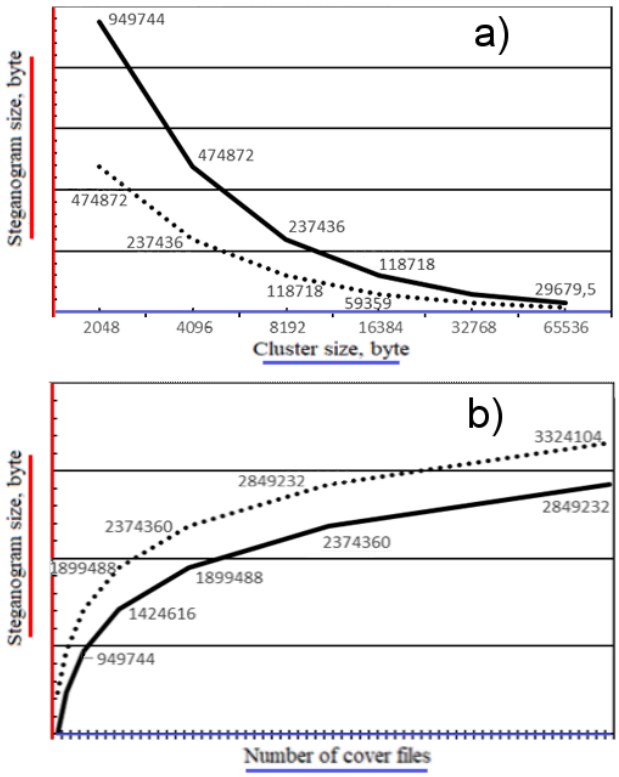


Fig. 2 – Dependence between:
- message length and size of a cluster (a), and count of cover files (b)

In fig. 3 shows the size of the steganograms depending on the number of cover files. There are comparing the results show that with equal input parameters, the proposed method allows hiding twice longer message than the basic method prototype. In addition, the improved method allows you to use only one cover file, as compared to the basic method. By comparing the computational complexity of the methods, it can be argued that the improved method requires twice the computational resources. For experimental research of the effectiveness of the methods, the program "*Stegano FAT*", FAT 32 file system on a JetFlash 350 Transcend® flash drive with a capacity of 8 GB, USB 2.0 connection interface and laptop Lenovo® Y510P with Windows® ver. 8.1 OS were used.

It should be noted that the actual execution time of hiding methods depends on both the hardware features of the data carriers and the algorithmic implementation. We will analyze the operation time of the methods, depending on the selected parameters: size of cluster; size of message; count of cover files; total size of the cover files.

To estimate the dependence of the time spent on the executing of the hiding and extracting messages methods to the cluster size, we set the message size is 100 bytes, the number of cover files – 2, the total size of the cover file – 7 MB.

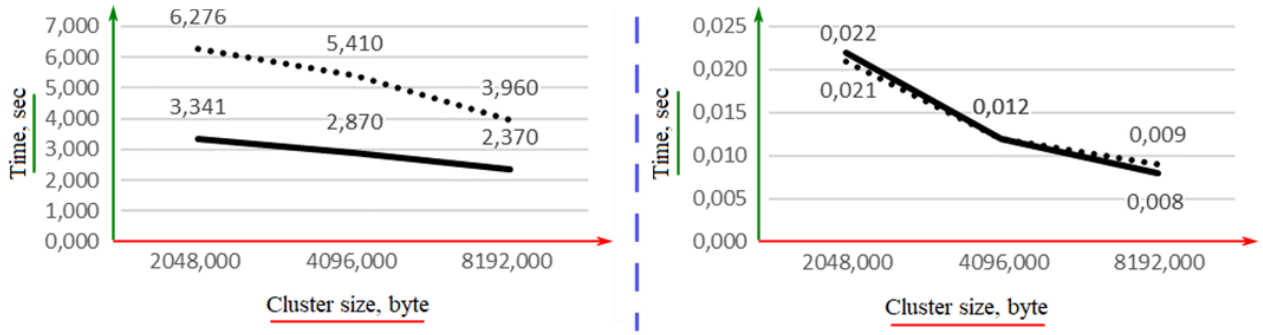


Fig. 3 – Effect of size of a cluster on spent time

We will change the cluster size in the range: 2048, 4096, 8192 bytes. The results of the experiment are summarized in Fig. 3. In this graph and the following ones, on the left are the results of hiding the message, on the right – when extracting. The dotted line shows the results of the modified method, the solid line shows the results of the basic method. As seen in Fig. 3, with increasing cluster size, the time of the concealment of the message is reduced. The improved method requires twice as much time to hide information in contrast to the based method, with the same configuration. The selected method does not affect the time of information retrieval. To estimate the dependence between the spent time on the hiding and extracting message and size of the message, we set the cluster size is 2048 bytes, the number of cover files – 2, the total size of the cover files 7 MB. We will change the size of the message: 100, 200, 400 bytes.

The results of the time spent analysis are shown in Fig. 4. As seen in Fig. 4 as the message size increases, the time to hide and extract the message increases. To estimate the dependence of the time spent on hiding and extracting the message to the number of cover files, we set the size of the cluster - 2048 bytes, the message size is 100 bytes, the total size of the cover files - 7 MB. We will change the number of cover files: 2,4,8. The results are shown in Fig. 5.

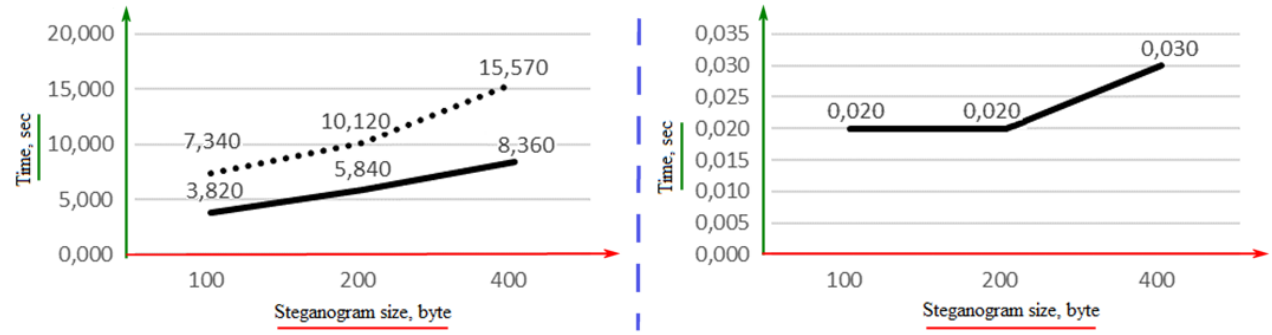


Fig. 4 – Effect of the size of the message on spent time

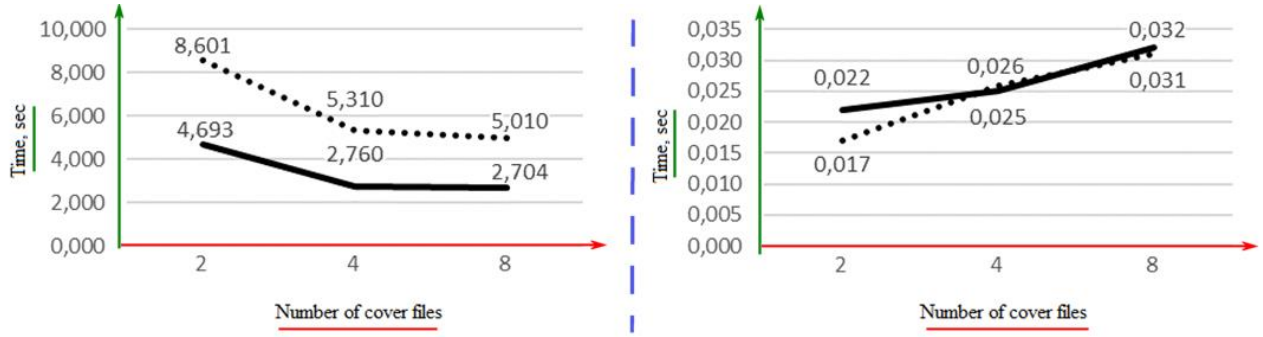


Fig. 5 – Effect of a count of cover files on spent time

As seen in Fig. 5, as the number of cover files increases, the time to hide the message - decreases. This is due to the fact that the number of information clusters, with an increase in the number of cover files, decreases, and accordingly increases the number of clusters that will be recorded without mixed. When a message is extracted, the time spent increases according to the number of cover files. To estimate the dependence of the time spent on the hiding and extracting of the message on the total size of the cover files, we fix the cluster size - 2048 bytes, the number of cover files - 2, size of the message is 100 bytes. We will change the total size of the cover files: 1.7, 3.5, 7 MB. The results are shown in Fig. 6. As seen in Fig. 6, when the total size of the cover files increasing, time to hiding and extracting the message increases.

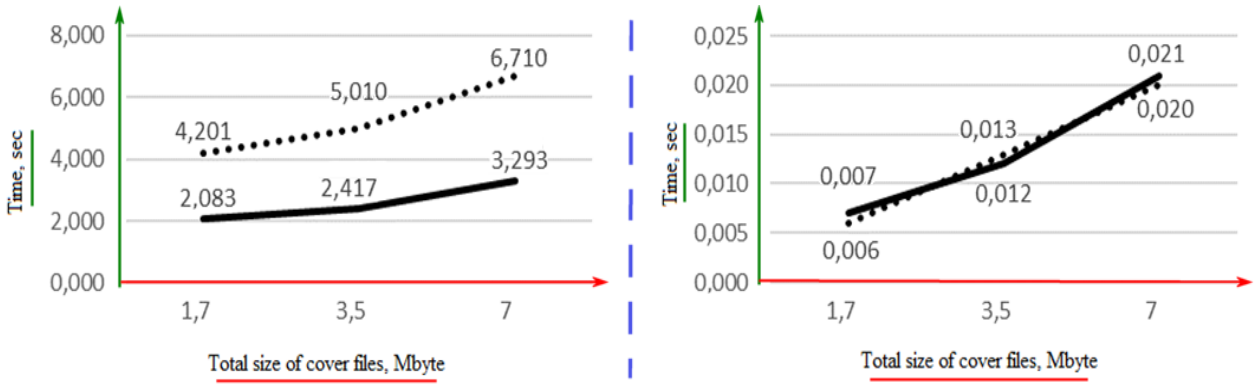


Fig. 6 – Effect of the total size of cover files on spent time

5 Conclusions

In this paper, we propose a new method, which, in contrast to the known besides regrouping clusters covering files additionally changes the order of alternating clusters in each of the cover files. It allows to further hide a specific information message, that is, increase the bandwidth of the hidden channel. The proposed method is implemented programmatically, the results of experimental research confirmed the adequacy of the theoretical conclusions and recommendations. There are results can be argued: the time of concealment and deletion of the message is largely influenced by the number of clusters over which we need to make a reposition; Extracting is performed much faster than concealing the message.

References

1. S. Katzenbeisser, F. A. Petitcolas. Information Hiding Techniques for Steganography and Digital Watermarking, Norwood, MA, USA: Artech House, 2000, 220 p.
2. F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn. "Information hiding-a survey," in Proc. of the IEEE, Vol. 87, No. 7, pp. 1062-1078, Jul 1999.
3. W. Mazurczyk, M. Smolarczyk, K. Szczypiorski. "Retransmission steganography and its detection", Soft Computing, Vol. 15, No. 3, pp. 505-515, 2011.
4. S. Nair, A. Kumar, A. Sur and S. Nandi. "Length based network steganography using UDP protocol". 2011 IEEE 3rd Int. Conference on Communication Software and Networks, Xi'an, 2011, pp. 726-730.
5. K. Ahsan and D. Kundur. "Practical data hiding in TCP lIP", In: ACM Workshop on Multimedia and Security, 2002, [On-line]. Internet: http://ee.tamu.edu/deepalpdf/ acm02.pdf
6. S. H. Sellke, C. Wang, S. Bagchi, and N. B. Shroff. "TCP/IP Timing Channels: Theory to Implementation", pp. 2204-2212, 2009.
7. V. Itier, W. Puech and A. G. Bors. "Cryptanalysis aspects in 3-D watermarking". 2014 IEEE International Conference on Image Processing (ICIP), Paris, 2014, pp. 4772-4776.
8. Yang, Qin, Liujie, Sun, and Wenju, Wang. "A robust watermarking scheme for 3D models based on encrypted holographic algorithm". Proceedings of 2015 International Conference on Intelligent Computing and Internet of Things, Harbin, 2015, pp. 85-89.
9. Z. Li, S. Beugnon, W. Puech, and A. G. Bors. "Rethinking the high capacity 3D steganography: Increasing its resistance to steganalysis". 2017 IEEE (ICIP), Beijing, 2017, pp. 510-414.
10. S. F. Liu, S. Pei, X. Y. Huang, and L. Tian. "File hiding based on FAT file system". 2009 IEEE International Symposium on IT in Medicine & Education, Jinan, 2009, pp. 1198-1201.

11. J. Davis, J. MacLean and D. Dampier. "Methods of Information Hiding and Detection in File Systems". 2010 Fifth IEEE International Workshop on Systematic Approaches to Digital Forensic Engineering, Oakland, CA, 2010, pp. 66-69.
12. H. Khan, M. Javed, S.A. Khayam, F. Mirza. “Designing a cluster-based covert channel to evade disk investigation and forensics”. Computers & Security, Vol. 30, Issue 1, January 2011. [On-line]. Internet: https://www.sciencedirect.com/science/article/pii/S016740481000088X
13. H. Khan, M. Javed, S.A. Khayam, F. Mirza. “Evading Disk Investigation and Forensics using a Cluster-Based Covert Channel”. National University of Science & Technology (NUST), Islamabad 44000, Pakistan. [On-line]. Internet: https://www.sigsac.org/ccs/CCS2009/pd/abstract_17.pdf
14. N. Morkevičius, G. Petraitis, A. Venčkauskas, J. Čeponis. “Covert Channel for Cluster-based File Systems Using Multiple Cover Files”. Information Technology and Control, 2013, Vol. 42, No.3. pp. 32. [On-line]. Internet: http://itc.ktu.lt/index.php/ITC/article/view/3328 .
15. L. Yang, P. Chen, G. Zhu, and L. Yu. "Repairing algorithm design for FAT file system in embedded system". 2011 International Conference on Consumer Electronics, Communications and Networks (CECNet), XianNing, 2011, pp. 3393-3396.
16. Z. Jinhai. "Research of embedded FAT file system". 2011 International Conference on Uncertainty Reasoning and Knowledge Engineering, Bali, 2011, pp. 44-47.
17. H. Zhao, X. Li, L. Chang, and X. Zang, "Fat File System Design and Research". 2015 International Conference on Network and Information Systems for Computers, Wuhan, 2015, pp. 568-571.

Рецензент: Сергій Толюпа, д.т.н., проф., Київський національний університет імені Т. Шевченка, м. Київ, Україна.
E-mail: tolupa@i.ua

Надійшло: Березень 2018.

Автори:
Олександр Кузнецов, д.т.н., проф., академік Академії наук прикладної радіоелектроніки, ХНУ імені В.Н. Каразіна, м. Харків, Україна.
E-mail: kuznetsov@karazin.ua

Кирил Шеханін, аспірант, ХНУ імені В.Н. Каразіна, м. Харків, Україна.
E-mail: kyryl.shekhanin@nure.ua

Андрій Колгатін, студент факультету комп’ютерних наук, Харківський національний університет імені В.Н. Каразіна, м. Харків, Україна.
E-mail: kolgatin-a@yandex.ua

Катерина Кузнецова, студентка факультету комп’ютерних наук, ХНУ імені В.Н. Каразіна, м. Харків, Україна.
E-mail: kate.kuznetsova.2000@gmail.com

Євген Деменко, студент факультету комп’ютерних наук, Харківський національний університет імені В.Н. Каразіна, м. Харків, Україна.
E-mail: demenjay@gmail.com

Приховування даних в файлової структурі.

Анотація. У статті досліджуються методи стеганографії, що приховують інформацію в структурі файлової системи. А саме, структура файлової системи FAT (таблиця розподілу файлів) і методи приховування інформаційних повідомлень, що засновані на зміні положення окремих кластерів файлів обкладинки. Пропонується новий метод, який, на відміну від відомих, змінює порядок слідування кластерів в кожному файлі обкладинки, що дозволяє додатково приховати інформаційне повідомлення, тобто збільшити ємність прихованого каналу. Підтверджено, що результати процедур приховування та вилучення даних в значній мірі залежать від кількості кластерів, з якими необхідно провести відповідні перетворення. Відзначено, що процедура вилучення виконується набагато швидше, ніж приховування повідомлення. Пропонований метод реалізований програмно, а результати експериментальних досліджень підтверджують правильність теоретичних висновків і рекомендацій.

Ключові слова: стеганографія; приховування інформаційних даних; файлова система.

Рецензент: Сергей Толюпа, д.т.н., проф., Киевский национальный университет имени Т. Шевченко, г. Киев, Украина.
E-mail: tolupa@i.ua

Поступила: Март 2018.

Авторы:
Александр Кузнецов, д.т.н., проф., академик Академии наук прикладной радиоэлектроники, ХНУ имени В.Н. Каразина, г. Харьков, Украина.

E-mail: kuznetsov@karazin.ua

Кирилл Шеханин, аспирант, ХНУ имени В. Н. Каразина, г. Харьков, Украина.
E-mail: kyryl.shekhanin@nure.ua

Андрей Колгатин, студент факультета компьютерных наук, ХНУ имени В.Н. Каразина, г. Харьков, Украина.

E-mail: kolgatin-a@yandex.ua

Екатерина Кузнецова, студентка факультета компьютерных наук, ХНУ имени В.Н. Каразина, г. Харьков, Украина.
E-mail: kate.kuznetsova.2000@gmail.com

Евгений Деменко, студент факультета компьютерных наук, ХНУ имени В.Н. Каразина, г. Харьков, Украина.
E-mail: demenjay@gmail.com

Скрытие данных в файловой структуре.

Аннотация. В статье исследуются методы стеганографии, скрывающие информацию в структуре файловой системы. А именно, структура файловой системы FAT (таблица распределения файлов) и методы скрытия информационных сообщений, которые основаны на изменении положения отдельных кластеров файлов обложки. Предлагается новый метод, который, в отличие от известных, изменяет порядок чередования кластеров в каждом файле обложки, что позволяет дополнительно скрыть информационное сообщение, то есть увеличить емкость скрытого канала. Подтверждено, что результаты процедур сокрытия и извлечения данных в значительной степени зависят от количества кластеров, с которыми необходимо провести соответствующие преобразования. Отмечено, что процедура извлечение выполняется гораздо быстрее, чем скрытие сообщения. Предлагаемый метод реализован программно, а результаты экспериментальных исследований подтверждают правильность теоретических выводов и рекомендаций.

Ключевые слова: стеганография; скрытие информационных данных; файловая система.

Статті пройшли внутрішнє та зовнішнє рецензування.

Наукове видання

КОМП'ЮТЕРНІ НАУКИ ТА КІБЕРБЕЗПЕКА

Випуск 1(9) 2018

Міжнародний електронний науково-теоретичний журнал

Англійською, українською, російською мовами

Комп'ютерне верстання – Єсіна М.В., Федоренко В.В.

61022, Харків, майдан Свободи, 6
Харківський національний університет імені В.Н. Каразіна

V. N. Karazin Kharkiv National University Publishing

CS&CS