Condensed Matter Physics 2008, Vol. 11, No 2(54), pp. 347–360

On the implementation of cryptoalgorithms based on algebraic graphs over some commutative rings

J.S.Kotorowicz*, V.A.Ustimenko†

University of Maria Curie-Sklodowska, Plac M.C. Sklodowkiej 1, 20–031 Lublin, Poland

Received January 31, 2008

The paper is devoted to computer implementation of some graph based stream ciphers. We compare the time performance of this new algorithm with fast, but no very secure RC4, and with DES. It turns out that some of new algorithms are faster than RC4. They satisfy the Madryga requirements, which is unusual for stream ciphers (like RC4). The software package with new encryption algorithms is ready for the demonstration.

Key words: *cryptography, directed graphs, stream ciphers*

PACS: *02.10.Hh, 02.10.Ox*

1. Introduction

We will study the properties of stream ciphers defined via finite automata corresponding to the family of algebraic graphs of high girth defined in [34]. The sequence of the graphs gives a well defined projective limit (infinite graphs) which is useful for theoretical studies of Turing machine corresponding to the graph based stream cipher. Algebraic nature of graphs implies the polynomiality of encryption scheme, but combinatorial properties make it possible to prove the absence of fixed points, and to establish that different keys produce distinct ciphertext. The transition functions of automaton related to graph form an arithmetical dynamic system in the sense of [33]. We consider the results of desynchronization of such infinite dynamical system, via applications of graph automorphisms and graph deformation, its effect on the security level and chaotical structure of cipher strings. The time evaluation of these algorithms defined via directed asymmetrical graphs compares well with the performance of fast but not very secure RC4, DES, algorithms based on simple graphs (symmetric anti-reflexive binary relations) developed during the recent ten years.

In section 2 we present the ideology of cryptography based on Extremal Graph Theory created by P. Erdös' and his school and some new results on extremal directed graphs, observe the results on cryptographical properties of graphs of high girth and implementations of algorithms based on automata related to such graphs.

In section 3, the reader can find basic cryptographical terminology.

Next section contains definitions of girth indicator and girth for directed graphs, concept of family of directed graphs of high girth, encryption automata related to members of such a family defined via special colouring of edges.

Section 5 is devoted to explicit constructions of algebraic families of graphs of large girth. For each commutative ring K we define two related families of algebraic directed graphs $RED(K)$ and $RDF(K)$ of large girth.

In section 5 and 6 we discuss the implementation of general stream cipher based of family of directed graphs of large girth $RDE_n(K)$, $n = 2, 3, \ldots$ in case of rings $K = Z_{2^k}$, $k \in \{8, 16, 32\}$. Graph $RDF_t(K)$, which is just a union of some connected components of $RDE_n(K)$, is useful in evaluating the girth indicator and the size of connected components of $RDE_n(K)$. We evaluate

*E-mail: jkotor@hektor.umcs.lublin.pl

†E-mail: vasyl@golem.umcs.lublin.pl

time performance and mixing properties of such algorithms and their modifications obtained via the desynchronization process.

2. On graphs of high girth and cryptoalgorithms

Studies of graphs with high girth had been motivated by problems of Networking ([1,7]). Since the well known work by R. Tanner [24], families of graphs of large girth have been instruments in Error Correction Theory (see [9,10,24] on the use of graphs of large girth for the creation of the so-called turbocodes). Recall that the girth is the length of the smallest cycle in the graph. The idea to use families of simple graphs of increasing girth in Cryptography had been explored in [11,25,27–33] and [34,37]. The encryption scheme for the "potentially infinite" text based on the family of graphs with special colouring of vertex set: the neighbours of each vertex are of different colours, there is a representative of each colour in the neighbourhood, the operator of taking the neigbour of a chosen color is a bijection. It is clear that the graphs have to be regular i.e. a size of the neighbourhood does not depend on the choice of vertex.

For this purpose, we identify the vertex of the graph with the plaintext. Encryption procedure corresponds to the chain of adjacent vertices (walk without consecutive edges) starting from the plaintext, the information on such chain being given by the sequence of corresponding colours (the password). We assume that the end of the chain is the ciphertext. It is easy to see that in the case when the length of the password is less than half of the girth, different passwords produce a distinct ciphertext corresponding to the same password and the ciphertext will be always different from the plaintext.

Notice that without loss of generality we can identify a set of colours with elements of commutative ring Z_n. We can attach the difference of colours for v_2 and v_1 to each directed edge (v_1, v_2) and convert the symmetric relation corresponding to the graph into finite automaton with the arbitrary initial state (plaintext). All states (vertices) of such automaton are accepting states. The above encryption procedure corresponds to some computation of this finite automaton. We will identify the graph and the corresponding binary relation.

For each $k \geqslant 3$ there is an infinite family of finite k-regular graphs G_i, $i = 1, 2, \dots$ of increasing order $|V_i|$ and increasing girth g_i (see, for instance [27,29]). In case of such a family with the appropriate colouring of its members as above we have a potentially infinite plainspace V_i, $i = 1, 2, \dots$ and a potentially infinite keyspace.

The ciphertext will be always different from the plaintext. If the minimal size of the connected component of each G_i is growing with i, then the encryption scheme is not a block cipher but a stream cipher. We can consider a more general encryption scheme defined by the sequence of k_i-regular graphs G_i, $i = 1, \dots$ of nondecreasing degree and increasing girth and order (see [27,37]).

The choice of simple graphs at the first stage (finite automata corresponding to symmetric binary relation) was motivated by classical Extremal Graph Theory. which deals with simple graphs only.

Let $e(G)$, $v = v(G)$ be the size (number of edges) and the order (number of vertices) of the graph G, respectively. Let $\mathrm{ex}(v, C_3, \dots, C_{2k})$ be the maximal size of the graph of the order of v without cycles $C_3, \dots, C_{2k}$. The following modification of Erdös' Even Circuit Theorem the reader can find in [6]:

$$\mathrm{ex}(v.C_3, \dots, C_{2k}) \leqslant cv^{1+1/k}, \tag{1}$$

where c is a positive constant independent of v. This bound is known to be sharp for $k = 2, 3$ and 5.

If the size of members G_i of the family of graphs of increasing girth g_i is close to the above bound ($g_i \geqslant C\log_{k_i}(v_i)$ for some positive constant C, which is the case of the so-called family of graphs of large girth) then the size of the plainspace and the maximal keyspace for the above encryption scheme are close to each other (see [28,29]).

In our encryption scheme we can "hide graphs up to isomorphism", i.e. take binary relation $\pi_i(G_i) = \{(u, v) | (\pi_i(v), \pi_i(u)) \in G_i\}$ instead of G_i, where π_i is some bijection on V_i. The sequence π_i is an invariant part of the key (password). Hiding graphs up to isomorphism is useful in case of graphs with the large automorphism group. It prevents the usage of automorphism of graphs during the attacks on the key.

An important feature of such encryption is the resistance to attacks, when adversary intercepts the pair plaintext – ciphertext (see [28]) , because the best algorithm of finding the pass between given vertices (by Dijkstra , see [6] and latest modifications) has a complexity $v \ln v$ where v is the order of the graph, i.e. size of the plainspace. The situation is similar to the checking of the primality of Fermat's numbers $2^{2^m}+1$: if the input is given by the string of binary digits, then the problem is polynomial, but if the input is given by just a parameter m, then the task is NP-complete.

We have an encryption scheme with the flexible length of the password (length of the chain). If graphs are connected and the length of password is not restricted, then we can convert each potentially infinite plaintext into the chosen string. In case of the so-called small world graphs we can do such a conversion "as fast as it is possible".

Finally, in the case of algebraic graphs in the sense of N.Biggs (see [2]), when the vertex set and neighbourhoods of each vertex are algebraic varieties over the same field (or ring) we have, in fact, a polynomial cryptography, because the operator of taking the neighbour of chosen colour is a polynomial map. So, such an encryption is easy to implement since software package with small memory is used (an example of the implementation of such public key encryption is in [30]). The first infinite family of algebraic graphs of large unbounded girth and arbitrary degree was constructed in [15](see [16] for the description of connected components). It was used in different software (different finite fields) packages developed via university projects at the University of South Pacific (Fiji Islands) [11,29,31], which serves for 11 remote island states within Pacific Ocean, Sultan Qaboos University (Oman) [25,32], University college of Cariboo (Canada, BC) [8], Ocanagan college, affiliated with the UBC (Canada), University of Kiev-Mohyla Academy(Ukraine), University of Maria Curie Sklodowska (Poland). The comparison of the first implementation of the algorithm (case field F_{127}) with another stream cipher private key algorithm ($RC4$) the reader can find in [8].

As we have already mentioned, the classical extreme graph theory deals with simple graphs. So, our first step was restricted to regular graphs of symmetric binary relations without loops. Let us assume that the graph is regular if for each vertex the numbers of inputs and outputs are equal to the same constant. The next step is reflected in [34] where the analog of P. Erdös' bound has been formulated for regular graphs of binary relations without loops and certain commutative diagrams. The analog of girth for directed graph is the so-called *girth indicator* (see section 4). The size of the graph is the total number of edges. Let $E_d(v)$ be the greatest size of a regular directed graph of the order of v the width girth indicator $> d$. The following analog of Erdös' Even Circuit Theorem has been formulated:

$$E_d(v) \leqslant v^{1+1/d}. \tag{2}$$

This bound turns out to be sharp not only for $d = 2, 3$ and 5 but for all $d \geqslant 2$ as well (see [36] for more general case of balanced directed graphs).

The paper [32] contains definitions of graphs of large girth (graphs with girth indicator d and size which is close to the above bound), small world graphs for this class of graphs, example of directed graphs based encryption. The bound and the related definition, the reader can find in section 4 below.

In the current paper we will discuss the implementation of algorithms which are based on families of nonsymmetric binary relation graphs. Instead of colouring the vertices we use special "rainbow-like colouring" of edges in the spirit of automata theory. In terms of such colouring we define graph based algorithms (subsection 4.2).

The practical advantage of directed graphs based cryptography in comparison with the previously used case of simple graphs is a much wider option to construct explicitly algebraic graphs over an arbitrarily chosen commutative ring K (section 5). Such K-theory has lead to very fast cryptoalgorithm (operation in $K = Z_{p^n}$ are much faster than in case of F_{p^n} for large n). In section 6 we compare the speed of some new algorithms with classical stream cipher $RC4$, used for the encryption of large data. In the last section we discuss some specific features of new encryption schemes.

3. Basic cryptographical terminology

Assume that an unencrypted message, *plaintext*, which can be image data, is a string of bytes. It is to be transformed into an encrypted string or *ciphertext*, by means of a cryptographic algorithm and a *key*: for the recipient to be able to read the message, encryption must be *invertible*.

Conventional wisdom holds that in order to defy easy decryption, a cryptographic algorithm should produce seeming chaos: that is, ciphertext should look and test random. In theory an eavesdropper should not be able to determine any significant information from an intercepted ciphertext. Broadly speaking, attacks to a cryptosystem fall into 2 categories: *passive attacks*, in which an adversary monitors the communication channel and *active attacks*, in which the adversary may transmit messages to obtain information (e.g. ciphertext of chosen plaintext).

Passive attacks are easier to mount, but yield less. Attackers hope to determine the plaintext from the ciphertext they capture; even more successful attacks will determine the key and thus comprise the whole set of messages.

An assumption first codified by Kerckhoffs in the nineteen century is that the algorithm is known and the security of algorithm rests entirely on the security of the key.

Cryptographers have been improving their algorithms to resist the following two major types of attacks:

i) *ciphertext only* – the adversary has access to the encrypted communications.

ii) *known plaintext* – the adversary has some plaintext and its corresponding ciphertext.

Nowadays the security of the plaintext rests on encryption algorithm (or private key algorithm), depending on the chosen key (password), which has good resistance to attacks of type (i), and algorithm for the key exchange with good resistance to attacks of type (ii) (public key algorithm).

The revolutionary classical result on private key algorithm was obtained by C. Shannon in the late 40th (see [12,13] or [23]). He constructed the so-called *absolutely secure* algorithms, whose keys and strings of random bits, at least as long as a message itself, achieve the seeming impossibility: an eavesdropper is not able to determine any significant information from the obtained ciphertext. The simplest classical example is the following one-time pad: if p_i is the i-th bit of the plaintext, k_i is the i-th bit of the key, and c_i is the first bit of the ciphertext, then $c_i = p_i + k_i$, where $+$ is exclusive or, often written XOR, and is simply addition modulo 2. One time pads must be used exactly once: if a key is ever reused, the system becomes highly vulnerable.

It is clear that the encryption scheme as above, like most private key algorithm, is irresistible to attacks of type (ii) – you need just subtract p_i from c_i and get the key.

4. Binary relations and related rainbow-like graphs, general symmetric algorithms

The missing theoretical definitions on directed graphs the reader can find in [22]. Let Φ be an irreflexive binary relation over the set V, i.e. $\Phi \in V \times V$ and for each v pair (v, v) is not the element of Φ.

We say that u is the neighbour of v if $(v, u) \in \Phi$. We use the term *binary relation graph* for the graph Γ of irreflexive binary relation ϕ over finite set V such that for each $v \in V$ sets $\{x|(x, v) \in \phi\}$ and $\{x|(v, x) \in \phi\}$ have same cardinality. It is a directed graph without loops and multiple edges, see [22] for more general definitions).

Let Γ be the graph of binary relation. The *pass* between vertices a and b is the sequence $a = x_0 \to x_1 \to \dots x_s = b$ of length s, where x_i, $i = 0, 1, \dots s$ are distinct vertices.

We say that the pair of passes $a = x_0 \to x_1 \to \dots \to x_s = b$, $s \geqslant 1$ and $a = y_0 \to y_1 \to \dots \to y_t = b$, $t \geqslant 1$ form an (s, t)- commutative diagram $O_{s,t}$ if $x_i \neq y_j$ for $0 < i < s$, $0 < j < t$. Without loss of generality we assume that $s \geqslant t$.

We refer to the number $\max(s, t)$ as the rank of $O_{s,t}$. It is $\geqslant 2$, because the graph does not contain multiple edges.

Notice, that the graph of antireflexive binary relation may have a directed cycle $O_s = O_{s,0}$: $v_0 \to v_1 \to \ldots v_{s-1} \to v_0$, where v_i, $i = 0, 1, \ldots, s-1$, $s \geqslant 2$ are distinct vertices.

We will count directed cycles as commutative diagrams.

In order to investigate the commutative diagrams we introduce *girth indicator* gi, which is the minimal value for $\max(s,t)$ for parameters s,t of commutative diagram $O_{s,t}$, $s+t \geqslant 3$. Notice that two vertices v and u at distance $<$ gi are connected by unique pass from u to v of length $<$ gi.

In case of symmetric binary relation $\mathrm{gi} = d$ implies that the girth of the graph is $2d$ or $2d-1$. It does not contain an even cycle $2d-2$. In general case $\mathrm{gi} = d$ implies that $g \geqslant d+1$. So, in the case of a family of graphs with unbounded girth indicator, the girth is also is unbounded. We also have $\mathrm{gi} \geqslant g/2$.

We assume that the *girth* $g(\Gamma)$ of the directed graph Γ with the girth indicator $d+1$ is $2d+1$ if it contains commutative diagram $O_{d+1,d}$. If there are no such diagrams we assume that $g(\Gamma)$ is $2d+2$.

In the case of symmetric irreflexive relations the above general definition of the girth agrees with the standard definition of the girth of simple graph, i.e the length of its minimal cycle.

We will use the term *the family of graphs of large girth* for the family of regular graphs Γ_i of degree k_i and the order v_i such that $\mathrm{gi}(\Gamma_i)$ is $\geqslant c\log_{k_i} v_i$, where c' is the constant independent of i. So the size of such graphs is quite close to the bound (2).

As it follows from the definition $g(\Gamma_i) \geqslant c'\log_{k_i}(v_i)$ for appropriate constant c'. So, it agrees with the well known definition for simple graphs.

4.1. Graphs with special colouring of vertices and edges, case of large girth

We shall use the term *the family of algebraic graphs* for the family of graphs $\Gamma(K)$, where K belongs to some infinite class F of commutative rings, such that the neighbourhood of each vertex of $\Gamma(K)$ and the vertex set itself are quasiprojective varieties over K of dimension $\geqslant 1$ (see [2,3] for the case of simple graphs).

Such a family can be treated as special Turing machine with the internal and external alphabet K.

We say that the graph Γ of binary relation Φ has a rainbow-like colouring over the set of colours C if for each v, $v \in V$ we have a colouring function ρ_v, which is a bijection from the neighbourhood $St(v)$ of v onto C, such that the operator $N_c(v)$ of taking the neighbour of v with colour c is the bijection of V onto V.

We say that the rainbow-like colouring ρ is invertible if there is a rainbow-like colouring of Φ^{-1} over C' such that ${N_c}^{-1} = N'_{c'}$ for some colour $c' \in C'$.

Example 1: Cayley graphs

Let G be the group and S be a subset of distinct generators, then the binary relation $\phi = \{(g_1, g_2) | g_i \in G, i = 1, 2, g_1 {g_2}^{-1} \in S$: admit the rainbow-like colouring $\rho(g_1, g_2) = g_1 {g_2}^{-1}$

This rainbow-like colouring is invertible because the inverse graph $\phi^{-1} = \{(g_2, g_1) | g_1 {g_2}^{-1} \in S\}$ admit the rainbow-like colouring $\rho'(g_2, g_1) = g_2 {g_1}^{-1} \in S^{-1}$.

Examples of Cayley graphs of large girth the reader can find in [18–20].

Example 2: Parallelotopic graphs and Latin squares

Let G be the graph with the colouring $\mu : V(G) \to C$ of the set of vertices $V(G)$ into colours from C such that the neighbourhood of each vertex looks like rainbow, i.e. consists of $|C|$ vertices of different colours. In case of pair (G, μ) we shall refer to G as *parallelotopic graph* with the local projection μ (see [27,28] and further references).

It is obvious that parallelotopic graphs are k-regular with $k = |C|$. If C' is a subset of C, then an induced subgraph $G^{C'}$ of G which consists of all vertices with colours from C' is also a parallelotopic graph. It is clear that the connected component of the parallelotopic graph is also a parallelotopic graph.

The *arc* of the graph G is a sequence of vertices $v_1, \ldots, v_k$ such that $v_i I v_{i+1}$ for $i = 1, \ldots, k-1$ and $v_i \neq v_{i+2}$ for $i = 1, \ldots, k-2$. If $v_1, \ldots, v_k$ is an arc of the parallelotopic graph (G, μ) then $\mu(v_i) \neq \mu(v_{i+2})$ for $i = 1, \ldots, k-2$.

Let $+$ be the Latin square defined on the set of colours C. Let us assume $\rho(u, v) = \mu(u) - \mu(v)$. The operator $N_c(u)$ of taking the neighbour of the color is invertible, ${N_c}^{-1} = N_{-c}$, where $-c$ is the opposite for c element in the Latin square. It means that ρ is invertible rainbow-like colouring.

We shall consider some examples of graphs with parallelotopic colouring in the section 8 and 9.

4.2. General symmetric algorithm

Let us consider the encryption algorithm corresponding to the graph Γ with the chosen invertible rainbow-like colouring of edges.

Let $\rho(u, v)$ be the colour of arrow $u \to v$. The set C is the totality of colours and $N_c(u)$ is the operator of taking the neighbour of u with the colour c.

The password is the string of colours $(c_1, c_2, \ldots, c_s)$ and the encryption procedure is the composition $N_{c_1} \times N_{c_2} \ldots N_{c_s}$ of bijective maps $N_{c_i} : V(\Gamma) \to V(\Gamma)$. So, if the plaintext $v \in V(\Gamma)$ is given, then the encryption procedure corresponds to the following chain in the graph: $x_0 = v \to x_1 = N_{c_1}(x_0) \to x_2 = N_{c_2}(x_1) \to \ldots \to x_s = N_{c_s}(x_{s-1}) = u$. The vertex u is the ciphertext.

Let $N'_{c'}(N_c(v)) = v$ for each $v \in V(\Gamma)$. The decryption procedure corresponds to the composition of maps $N'_{c'_s}, N'_{c'_{s-1}}, \ldots, N'_{c'_1}$. The above scheme gives a symmetric encryption algorithm with flexible length of the password (key). Let $A(\Gamma, \rho, s)$ be the above encryption scheme. The following statement is immediate corollary from definitions.

Lemma 1 *Let Γ be the invertible rainbow-like graph of girth g and $A(\Gamma, \rho, s)$ be the above encryption scheme for $s <$ (gi). Then different passwords produce distinct ciphertexts, plaintext and the corresponding ciphertext are different.*

5. The incidence structures defined over commutative rings

E. Moore [21] used the term *tactical configuration* of the order (s, t) for biregular bipartite simple graphs with bidegrees $s + 1$ and $r + 1$. It corresponds to the incidence structure with the point set P, line set L and symmetric incidence relation I. Its size can be computed as $|P|(s + 1)$ or $|L|(t + 1)$.

Let $F = \{(p, l) | p \in P, l \in L, pIl\}$ be the totality of flags for the tactical configuration with partition sets P (point set) and L (line set) and incidence relation I. We define the following irreflexive binary relation ϕ on the set F:

$((l_1, p_1), (l_2, p_2)) \in \phi$ if and only if $p_1 I l_2$, $p_1 \neq p_2$ and $l_1 \neq l_2$.

Let $F(I)$ be the binary relation graph corresponding to ϕ. The order of $F(I)$ is $|P|(s + 1)$ (or $|L|(t + 1)$ We refer to it as *directed flag graph* of I.

Let (P, L, I) be the incidence structure corresponding to regular tactical configuration of the order t.

Let $F_1 = \{(l, p) | l \in L, p \in P, lIp\}$ and $F_2 = \{[l, p] | l \in L, p \in P, lIp\}$ be two copies of the totality of flags for (P, L, I). Brackets and parenthesis allow us to distinguish elements from F_1 and F_2. Let $DF(I)$ be the directed graph (double directed flag graph) on the disjoint union of F_1 with F_2 defined by the following rules:

$(l_1, p_1) \to [l_2, p_2]$ if and only if $p_1 = p_2$ and $l_1 \neq l_2$,

$[l_2, p_2] \to (l_1, p_1)$ if and only if $l_1 = l_2$ and $p_1 \neq p_2$.

We will define hereinafter the family of graphs $D(k, K)$, where $k > 2$ is positive integer and K is a commutative ring. Such graphs have been considered in [15] for the case $K = F_q$ (some examples are in [14]).

Let P and L be two copies of Cartesian power K^N, where K is the commutative ring and N is the set of positive integer numbers. Elements of P will be called *points* and those of L *lines*.

To distinguish the points from the lines we use parentheses and brackets: If $x \in V$, then $(x) \in P$ and $[x] \in L$. It will also be advantageous to adopt the notation for co-ordinates of points and lines introduced in [17] for the case of general commutative ring K:

$$(p) = \left(p_{0,1}, p_{1,1}, p_{1,2}, p_{2,1}, p_{2,2}, p'_{2,2}, p_{2,3}, \ldots, p_{i,i}, p'_{i,i}, p_{i,i+1}, p_{i+1,i}, \ldots\right),$$

$$[l] = \left[l_{1,0}, l_{1,1}, l_{1,2}, l_{2,1}, l_{2,2}, l'_{2,2}, l_{2,3}, \ldots, l_{i,i}, l'_{i,i}, l_{i,i+1}, l_{i+1,i}, \ldots\right].$$

The elements of P and L can be thought of as infinite ordered tuples of elements from K, such that only finite number of components are different from zero.

We now define an incidence structure (P, L, I) as follows. We say that the point (p) is incident with the line $[l]$, and we write $(p)I[l]$, if the following relations between their co-ordinates hold:

$$\begin{aligned} l_{i,i} - p_{i,i} &= l_{1,0}p_{i-1,i}, \qquad l'_{i,i} - p'_{i,i} = l_{i,i-1}p_{0,1} \\ l_{i,i+1} - p_{i,i+1} &= l_{i,i}p_{0,1}, \qquad l_{i+1,i} - p_{i+1,i} = l_{1,0}p'_{i,i} \end{aligned} \tag{3}$$

(these four relations are defined for $i \geqslant 1$, $p'_{1,1} = p_{1,1}$, $l'_{1,1} = l_{1,1}$). We denote this incidence structure (P, L, I) as $D(K)$. We identify it with the bipartite *incidence graph* of (P, L, I), which has the vertex set $P \cup L$ and the edge set consisting of all pairs $\{(p), [l]\}$ for which $(p)I[l]$.

For each positive integer $k \geqslant 2$ we obtain an incidence structure (P_k, L_k, I_k) as follows. First, P_k and L_k are obtained from P and L, respectively, by simply projecting each vector onto its k initial coordinates with respect to the above order. The incidence I_k is then defined by imposing the first $k-1$ incidence equations and ignoring all the others. The incidence graph corresponding to the structure (P_k, L_k, I_k) is denoted by $D(k, K)$.

To facilitate the notation in the future results, it will be convenient for us to define $p_{-1,0} = l_{0,-1} = p_{1,0} = l_{0,1} = 0$, $p_{0,0} = l_{0,0} = -1$, $p'_{0,0} = l'_{0,0} = -1$, and to assume that (3) are defined for $i \geqslant 0$.

Notice that for $i = 0$, the four conditions (3) are satisfied by every point and line, and, for $i = 1$, the first two equations coincide and give $l_{1,1} - p_{1,1} = l_{1,0}p_{0,1}$.

The incidence relation motivated by the linear interpretation of Lie geometries in terms of their Lie algebras [26]. Let us define the "root subgroups" U_α, where the "root" α belongs to the root system Root $= \{(1,0), (0,1), (1,1), (1,2), (2,1), (2,2), (2,2)' \ldots, (i,i), (i,i)', (i,i+1), (i+1,i) \ldots\}$. The "root system above" contains all real and imaginary roots of the Kac-Moody Lie Algebra $\tilde{A}_1$ with the symmetric Cartan matrix. We just double imaginary roots (i,i) by introducing $(i,i)'$.

Remark. For $K = F_q$ the following statement had been formulated in [17].

Let $k \geqslant 6$, $t = [(k+2)/4]$, and let $u = (u_\alpha, u_{11}, \cdots, u_{tt}, u'_{tt}, u_{t,t+1}, u_{t+1,t}, \cdots)$ be a vertex of $\mathrm{D}(k, K)$ ($\alpha \in \{(1,0), (0,1)\}$. It does not matter whether u is a point or a line). For every r, $2 \leqslant r \leqslant t$, let

$$a_r = a_r(u) = \sum_{i=0,r} (u_{ii}u'_{r-i,r-i} - u_{i,i+1}u_{r-i,r-i-1}),$$

and

$$a = a(u) = (a_2, a_3, \cdots, a_t).$$

Proposition 2 *(i) The classes of equivalence relation $\tau = \{(u,v) | a(u) = a(v)\}$ are connected components of graph $D(n, K)$, where $n \geqslant 2$ and K is the ring with unity of characteristic > 2.*

(2i) For any $t-1$ ring elements $x_i \in K$), $2 \leqslant t \leqslant [(k+2)/4]$, there exists a vertex v of $\mathrm{D}(k, K)$ *for which*

$$a(v) = (x_2, \ldots, x_t) = (x).$$

(3i) The equivalence class C for the equivalence relation τ on the set $K^n \cup K^n$ is isomorphic to the affine variety $K^t \cup K^t$, $t = [4/3n] + 1$ for $n = 0, 2, 3 \bmod 4$, $t = [4/3n] + 2$ for $n = 1 \bmod 4$.

Remark. Let K be the general commutative ring and C be the equivalence class on τ on the vertex set D(K) (D(n, K)). Then the induced subgraph, with the vertex set C is the union of several connected components of D(K) (D(n, K), respectively).

Without loss of generality we may assume that for the vertex v of $C(n, K)$ satisfying $a_2(v) = 0, \ldots, a_t(v) = 0$. We can find the values of components $v'_{i,i}$ from this system of equations and eliminate them. Thus we can identify P and L with elements of K^t, where $t = [3/4n] + 1$ for $n = 0, 2, 3 \bmod 4$, and $t = [3/4n] + 2$ for $n = 1 \bmod 4$.

We shall use notation $C(t, K)$ ($C(K)$) for the induced subgraph of $D(n, K)$ with the vertex set C.

Remark.

If $K = F_q$, q is odd, then the graph $C(t, k)$ coincides with the connected component $CD(n, q)$ of the graph $D(n, q)$ (see [17]), graph $C(F_q)$ is a q-regular tree. In other cases the question on the connectivity of $C(t, K)$ is open. It is clear that $g(C(t, F_q))$ is $\geqslant 2[2t/3]+4$. For each positive integer $k \geqslant 2$ we consider the *standard* graph homomorphism ϕ_k of

(P_k, L_k, I_k) onto $(P_{k-1}, L_{k-1}, I_{k-1})$ defined L_k by simply projecting each vector from P_k and L_k onto its $k-1$ initial coordinates with respect to the above order.

Proposition 3 *Projective limit of graphs $D(n, K)$ (graphs $C(t, K)$, $CD(n, K)$) with respect to standard morphisms of $D(n+1, K)$ onto $D(n, K)$ (their restrictions on induced subgraphs) is equal to $D(K)$ $(C(K), respectively)$.*

If K is an integrity domain, then $D(K)$ and $CD(K)$ are forests. Let C be the connected component, i.e tree.

Let us consider the directed flag graph $F(t, K)$ of the tactical configuration $C(t, K)$. Let $E(n, K)$ be the directed flag graph of bipartite graph $D(n, K)$. We can consider the symbolic invertible rainbow-like colouring $\rho(f_1, f_2)$ of $F(t, K)$ (or $E(n, K)$) defined on the colour set $K^* \times K^*$ by the following rule:

Let $f_1 = ([l^1], (p^1))$, $f_2 = ([l^2], (p^2))$ form the arrow in $F(t, K)$. So, $[l^2]I(p^1)$. We assume that $\rho(f_1, f_2) = (l^1_{1,0} - l^2_{1,0}, p^1_{0,1} - l^2_{0,1})$.

If K is finite, then the cardinality of the colour set is $(|K| - 1)^2$. Let Reg$K$ be the totality of regular elements, i.e. not zero divisors. Let us delete all arrows with colour (x, y), where one of the elements x and y is a zero divisor in the graph $F(t, K)$ ($E(n, K)$, respectively). New graph $RF(t, K)$ ($RE(n, K)$, respectively) is a symbolic rainbow-like graph over the set of colours RegK^2.

The following statement is proven in [37].

Theorem 4 *The girth indicator* gi *of the algebraic rainbow-like graph $RF(t, K)$ is $\geqslant 1/3t$.*

Corollary 5 *Let K be a finite commutative ring such that $k = |\mathrm{Reg}K| \geqslant 2$. Then graphs $RF(t, K)$, $t = 2, 3, \ldots$ form the family of algebraic rainbow-like graphs of large girth of bounded degree.*

The graph $RF(t, K)$ is an induced subgraph in $RE(n, K)$. So we get the following statement.

Corollary 6 *The girth indicator* gi *of the algebraic rainbow-like graph $RE(n, K)$ is $\geqslant 1/4n + 1$.If $k = |\mathrm{Reg}K| \geqslant 2$, then graphs $RE(n, K)$, $n = 2, 3, \ldots$ form the family of algebraic rainbow-like graphs of large girth of bounded degree.*

Let us consider the double directed flag graph $DF(t, K)$ of the tactical configuration $C(t, K)$. Let $DE(n, K)$ be the double directed graph of the bipartite graph $D(n, K)$. Remember, that we have the arc e of type $(l^1, p^1) \to [l^2, p^2]$ if and only if $p^1 = p^2$ and $l^1 \neq l^2$. Let us assume that the colour $\rho(e)$ of arc e is $l^1_{1,0} - l^2_{1,0}$.

Recall, that we have the arc e' of type $[l^2, p^2] \to (l^1, p^1)$ if and only if $l^1 = l^2$ and $p^1 \neq p^2$. let us assume that the colour $\rho(e')$ of arc e' is $p^1_{1,0} - p^2_{1,0}$.

If K is finite, then the cardinality of the colour set is $(|K|-1)$. Let Reg$K$ be the totality of regular elements, i.e. not zero divisors. Let us delete all arrows with colour, which is a zero divisor. New graph $RDF(t,K)$ is an algebraic rainbow-like graph over the set of colours (RegK)

Similarly to the previous theorem we formulate the following proposition.

Theorem 7 *The girth indicator* gi *of the algebraic rainbow-like graph $RDF(t,K)$ ($RDE(n,K)$) is $\geqslant 2/3t$ ($\geqslant n/2+1$, respectively).*

Corollary 8 *Let K be a finite such that $k = |\mathrm{Reg}K| \geqslant 2$. Then graphs $RDF(t,K)$, $t = 2,3,\ldots$ ($RDE(n,K)$, $n = 2,3\ldots$) form the family of algebraic rainbow-like graphs of large girth of bounded degree.*

Standard homomorphism ϕ_n of $D(n,K)$ onto $D(n-1,K)$ induces type preserving standard graph homomorphism of $RDE(n,K)$ onto $RDE(n-1,K)$) ($RDF(n,K)$ onto $RDF(n-1,K)$). So graphs $RDE(n-1,K)$ ($RDF(n,K)$) form the folder of rainbow-like graphs (see section 2).

Projective limit of graphs $RDE(n,K)$ (graphs $RDF(n,K)$) with respect to standard homomorphisms is well defined.

6. Time evaluation

We have implemented the computer application, which uses a family of graphs $RDE(n,K)$ for *private key* cryptography. To achieve high speed property, commutative ring $K = Z_{2^k}$, $k \in \{8,16,32\}$, with operations $+, \times$ modulo 2^k. Parameter n stands for the length of plaintext (input data) and the length of ciphertext. We mark by $G1$ the algorithm with $k=8$, by $G2$ the algorithm with $k=16$, and by $G4$ the algorithm with $k=32$. So $Gi, i \in 1,2,4$ denotes the number of bytes used in the alphabet (and the size of 1 character in the string).

The alphabet for password is the same K as for the plaintext. For an encryption we use the scheme presented in section (4.1). The colour of vertex is its first coordinate.

If u is the vertex, $p(u)$ is the colour of this vertex, and α is the character of password, then the next vertex in the encryption path v has the colour $p(v) = p(u) + \alpha$. All the next coordinates of v are computed from (3) set of equations.

All the tess were run on computer with parameters:

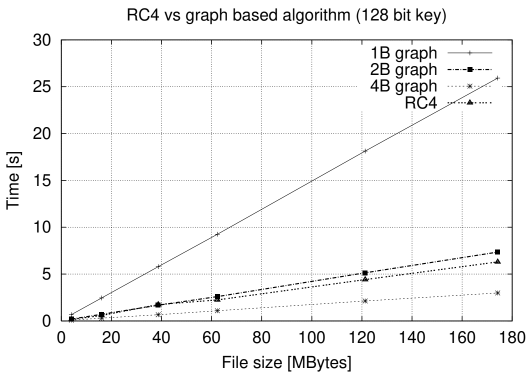
- AMD Athlon 1.46 GHz processor
- 1 GB RAM memory
- Windows XP operating system.

The program was written in Java language. Well known algorithms RC4 and DES which were used for comparison have been taken from Java standard library for cryptography purposes – *javax.crypto*.

6.1. Our algorithm compared with RC4

RC4 is a well known and widely used stream cipher algorithm. Protocols SSL (to protect Internet traffic) and WEP (to secure wireless networks) uses it as an option. Nowadays RC4 is not secure enough and not recommended for use in a new system. Anyway we chose it for the sake of comparison, because of its popularity and high speed.

RC4 is not dependent on password length in terms of complexity, while our algorithm is dependent on it. Longer password makes us do more steps between vertices of a graph. So, for fair comparison we have used fixed password length equal to the suggested upper bound for RC4 (16 Bytes).



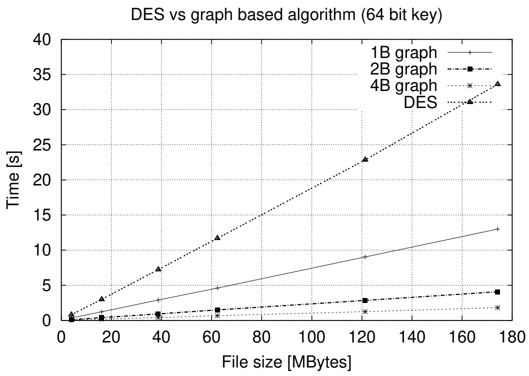
File [MB]	RC4 [s]	G1 [s]	G2 [s]	G4 [s]
4	0.15	0.67	0.19	0.08
16.1	0.58	2.45	0.71	0.30
38.7	1.75	5.79	1.68	0.66
62.3	2.24	9.25	2.60	1.09
121.3	4.41	18.13	5.14	2.13
174.2	6.30	25.92	7.35	2.98

Figure 1. RC4 vs high girth graph based algorithm (128 bit password).

6.2. Comparison with DES

In the next test we have compared our algorithm with popular block cipher DES (Data Encryption Standard). DES is more complicated and has better cryptographical properties than RC4, but it is much slower.

The version of DES implemented in Java library uses 64 bit password and makes from it 56 bit key (due to documentation). In our comparison (see figure (2)) we used the password of the same length.



File [MB]	DES [s]	G1 [s]	G2 [s]	G4 [s]
4	0.81	0.35	0.11	0.05
16.1	2.99	1.23	0.40	0.18
38.7	7.24	2.90	0.92	0.41
62.3	11.69	4.60	1.49	0.68
121.3	22.85	9.03	2.85	1.25
174.2	33.60	13.00	4.08	1.82

Figure 2. DES vs high girth graph based algorithm, 64 bit password.

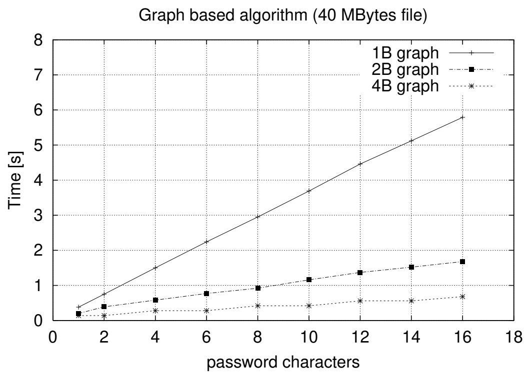
6.3. Linearity from password length

It is easy to understand that with the fixed size of the plaintext, our algorithm depends linearly on the password length. Each step of algorithm (taking the neigbour of the chosen colour) has a fixed complexity, and the number of such steps depends on the number of characters in the password.

Figure (3) illustrates this property, and shows the advantage of using bigger alphabet, but a less number of operations. Algorithm "G4", using up-to-date 32 bit arithmetics (with automatic modulo operations) behaves over 8 times faster than "G1" (8 bit arithmetics).

7. Statistics related to mixing properties

In our cryptographical scheme, different passwords produce different ciphertexts with fixed plaintext. On the other hand, when we fix the password, different plaintexts produce different



Pass [B]	G1 [s]	G2 [s]	G4 [s]
1	0.38	0.20	0.14
2	0.75	0.39	0.14
4	1.50	0.58	0.28
6	2.24	0.77	0.28
8	2.95	0.92	0.42
10	3.69	1.16	0.42
12	4.46	1.37	0.56
14	5.12	1.52	0.56
16	5.79	1.68	0.68

Figure 3. Fixed file size (40 MB), comparison of our 3 algorithms.

cypertexts. Good cryptographical systems should ensure this difference to be big in terms of the number of characters changed, looking as "randomly" as possible. These demands are known in literature as *Madryga requirements*. There are more postulates for a good cryptosystem formulated by Madryga, but here we will concentrate on the two mentioned ones.

RC4 algorithm, as most elder stream ciphers, possesses the property at which, in case of fixed password, the change of one element of the plaintext leads to the change of one corresponding element in the ciphertext. Such algorithms are not secure against the *plaintext-ciphertext* attacks.

Our basic algorithm, based on the paths in graphs from the family $RDE(n, K)$, behaves similarly to RC4: the change of one element in the plaintext leads to the change of only a few elements in the ciphertext.

In order to correct this property, we can combine the algorithm with some fast, sparse matrix operations:

1. *Desynchronization* of the graph by the automorphism.
   Let $\bar{a} = (a_1, a_2, \ldots, a_m), (a_i \in Z_{2^k})$ be the password and N_{a_i} be one step of algorithm (passing from one vertex to another using a_i element of password). We can denote our encryption algorithm as
   $$E_{\bar{a}} = N_{a_1} N_{a_2} \ldots N_{a_m}.$$
   Desynchronization can be described as:
   $$\mathbf{A} N_{a_1} N_{a_2} \ldots N_{a_m} \mathbf{A}^{-1} = \mathbf{A} N_{a_1} \mathbf{A}^{-1} \mathbf{A} N_{a_2} \mathbf{A}^{-1} \ldots \mathbf{A} N_{a_m} \mathbf{A}^{-1},$$
   where $\mathbf{A}$ is some bijection. All the properties of $E_{\bar{a}}$ that are of interest to us are preserved.

2. *Deformation* of the graph.
   With the above notation for the deformation we use two bijections $\mathbf{A}$ and $\mathbf{B}$, changing $E_{\bar{a}}$ into $\mathbf{A}E_{\bar{a}}\mathbf{B}$. The property that different passwords lead to different ciphertexts is preserved, but there can happen the situation, that for the plaintext vector $\bar{x}$ the corresponding ciphertext, $\mathbf{A}E_{\bar{a}}\mathbf{B}(\bar{x})$ coinsides with $\bar{x}$. Anyway the probability of such an event is $1/|V|$, where V is the plainspace. It is very close to zero.

We chose the bijection A as sparse affine transformation. Its complexity is $O(n)$. Our test shows, that a properly chosen upper-triangular matrix $\mathbf{A}_n$ used for desynchronization gives about 98.5% difference between the ciphertexts, when changing only 1 element of the plaintext (we use index n, because the size of the $\mathbf{A}$ depends on the size of the plaintext). Table (1) shows the extra time spent by all 3 versions of our algorithm on the operation $\mathbf{A}_n$.

If instead of desynchronization as above we apply the deformation with $\mathbf{B} = \mathbf{I}$ (identity map) and the same $\mathbf{A}$, the speed of computation will be twice better and mixing properties remain the same.

Table 1. Time growth from mixing property $\mathbf{A}_n E_{\bar{a}} \mathbf{A}_n^{-1}$ for chosen operator $\mathbf{A}_n$.

File [MB]	G1 [s]	G2 [s]	G4 [s]
4	0.04	0.02	0.01
16.1	0.12	0.10	0.08
38.7	0.32	0.24	0.20
62.3	0.50	0.40	0.30
121.3	0.96	0.76	0.60
174.2	1.39	0.96	0.74

The second Madryga requirement mentioned above (the effect of the change of one character from the key) can be stated as follows: for short passwords (1B) the percentage of the change within the cipherstring is about 92%, and for longer passwords it is up to 96%.

References

1. Bien F., Constructions of telephone networks by group representations. Notices Amer. Mah. Soc., 1989, **36**, 5–22.
2. Biggs N. Algebraic Graph Theory, (2nd ed). University Press, Cambridge, 1993.
3. Biggs N.L., Graphs with large girth, Ars Combinatoria, 1988, **25C**, 73–80.
4. Bollobás B. Extremal Graph Theory. Academic Press, London, 1978.
5. Bollobás B. Random Graphs. Academic Press, London, 1985.
6. Dijkstra E.,A note on two problems in connection with graphs, Num. Math., 1959, **1**, 269–271.
7. Erdös' P., Sachs H., Regulare Graphen gegebener Taillenweite mit minimaler Krotenzahl, Wiss. Z. Univ. Halle Martin Luther, Univ. Halle-Wittenberg, Math. Natur. Reine, 1963, **12**, 251–257.
8. Govorov M., Khmelevsky Y., Ustimenko V., Khorev A. Security for GIS N-tier Architecture, 11th International Symposium on Spatial Data Handling Fisher, Peter F. (Ed.) Springer-Verlag. 71, 2005.
9. Guinand P., Lodge J. Tanner Type Codes Arising from Large Girth Graphs, Proceedings of the 1997 Canadian Workshop on Information Theory (CWIT '97), Toronto, Ontario, Canada, pp. 5–7, June 3–6, 1997.
10. Guinand P., Lodge J. Graph Theoretic Construction of Generalized Product Codes, Proceedings of the 1997 IEEE International Symposium on Information Theory (ISIT '97), Ulm, Germany, p. 111, June 29–July 4, 1997.
11. Khmelevsky Yu., Ustimenko V.A., Practical aspects of the Informational Systems reengineering, The South Pacific Journal of Natural Science, volume 21, 2003, www.usp.ac.fj(spjns).
12. Koblitz N. A Course in Number Theory and Cryptography, Second Edition, Springer, 1994, 237 p.
13. Koblitz N. Algebraic aspects of Cryptography, in Algorithms and Computations in Mathematics, v. 3, Springer, 1998.
14. Lazebnik F., Ustimenko V.A., New Examples of graphs without small cycles and of large size, Europ. J. of Combinatorics, 1993, **14**, 445–460.
15. Lazebnik F., Ustimenko V., Explicit construction of graphs with an arbitrary large girth and of large size, Discrete Appl. Math., 1995, **60**, 275–284.
16. Lazebnik F., Ustimenko V.A., Woldar A.J. A New Series of Dense Graphs of High Girth, Bull (New Series) of AMS, 1995, **32**, No. 1, 73–79.
17. Lazebnik F., Ustimenko V.A., Woldar A.J., A characterization of the components of graphs $D(k, q)$, Discrete Mathematics, 1996, **157**, 271–283.
18. Margulis G.A., Explicit construction of graphs without short cycles and low density codes, Combinatorica, 1982, **2**, 71–78.
19. Margulis G., Explicit group-theoretical constructions of combinatorial schemes and their application to desighn of expanders and concentrators, Probl. Peredachi Informatsii., 24, N1, 51–60. English translation publ. Journal of Problems of Information transmission, 1988, 39–46.
20. Margulis M., Arithmetic groups and graphs without short cycles, 6th Intern. Symp. on Information Theory, Tashkent, abstracts, vol. 1, 1984, pp. 123–125 (in Russion).

21. Moore E.H., Tactical Memoranda, Amer. J. Math., 1886, **18**, 264–303.
22. Ore R. Graph Theory. London, 1974.
23. Seberry J., Pieprzyk J. Cryptography: An Introducion to Computer Security. Prentice Hall, 1989, 379 p.
24. Tanner R.M. A recursive approach to low density codes, IEEE Trans. on Info Th., IT, 27(5):533–547, Sept. 1984.
25. Touzene A., Ustimenko V., Graph Based Private Key Crypto System, International Journal on Computer Research, Nova Science Publisher, 2006, **13**, Iss. 4, 12.
26. Ustimenko V.A., Linear interpretation of Chevalley group flag geometries, Ukrainian Math. J., 1991, **43**, No. 7–8, 1055–1060 (in Russian).
27. Ustimenko V.A., Coordinatisation of regular tree and its quotients, in "Voronoi's impact on modern science", eds P. Engel and H. Syta, book 2, National Acad. of Sci, Institute of Matematics, 1998, 228p.
28. Ustimenko V., Graphs with Special Arcs and Cryptography, Acta Applicandae Mathematicae, 2002, **74**, No. 2, 117–153.
29. Ustimenko V., CRYPTIM: Graphs as tools for symmetric encryption, In Lecture Notes in Comput. Sci., 2227, Springer, New York, 2001.
30. Ustimenko V., Maximality of affine group and hidden graph cryptsystems, Journal of Algebra and Discrete Mathematics, October, 2004, **10**, 51–65.
31. Ustimenko V.A., Sharma D., CRYPTIM: system to encrypt text and image data, Proceedings of International ICSC Congress on Intelligent Systems 2000, Wollongong, 2001, 11pp.
32. Ustimenko V., Touzene A., CRYPTALL:system to encrypt all types of data, Notices of Kiev-Mohyla Academy, v 23, June , 2004, pp. 12–15.
33. Ustimenko V.A., Linguistic Dynamical Systems, Graphs of Large Girth and Cryptography, Journal of Mathematical Sciences, 2007, **140**, No. 3, 412–434.
34. Ustimenko V., On the extremal graph theory for directed graphs and its cryptographical applications, In: Shaska T., Huffman W.C., D. Joener and V. Ustimenko, Advances in Coding Theory and Cryptography, Series on Coding Theory and Cryptology, 2007, **3**, 181–200.
35. Ustimenko V.A., Algebraic small world graphs of large girth and related groupd, Proceedings of the international conferences "Infinite particle systems, Complex systems theory and its application", Kazimerz Dolny, Poland, 2005–2006 (to appear).
36. Ustimenko V.A., On the extremal binary relation graphs of high girth, Proceedings of the international conferences "Infinite particle systems, Complex systems theory and its application", Kazimerz Dolny, Poland, 2005–2006 (to appear).
37. Ustimenko V.A., On graph based cryptography and symbolic computations, Serdica J. of Computing, 2007, **1**, 131–156.

До застосування алгоритмів шифрування, що базуються на алгебраїчних графах

Й.С.Которович, В.А.Устименко

Університет ім. Марії Кюрі-Склодовської, Люблін, Польща

Отримано 31 січня 2008 р.

Стаття присвячена комп'ютерному застосуванню деяких потокових алгоритмів шифрування, що базуються на графах. Ми порівнюємо швидкість нового алгоритму з швидким, але не дуже безпечним алгоритмом RC4 та DES. Ці алгоритми задовільняють вимоги Мадриги, що є нетиповим для потокових алгоритмів (типу RC4). Пакет комп'ютерних програм на основі нового алгоритму є готовим для демонстрації.

Ключові слова: *шифрування, напрямлені графи, потокові алгоритми шифрування*

PACS: *02.10.Hh, 02.10.Ox*