ISBN 978-83-60810-27-9
ISSN 1896-7094

On the implementation of public keys algorithms based on algebraic graphs over finite commutative rings

Michał Klisowski
Maria Curie-Sklodowska University,
Institute of Mathematics,
pl. M. Curie-Skłodowskiej 1, 20-031 Lublin, Poland
Email: mklisow@hektor.umcs.lublin.pl

Vasyl Ustimenko
Maria Curie-Sklodowska University,
Institute of Mathematics,
pl. M. Curie-Skłodowskiej 1, 20-031 Lublin, Poland
Email: vasyl@hektor.umcs.lublin.pl

***Abstract*—We will consider** balanced **directed graphs, i.e., graphs of binary relations, for which the number of inputs and number of outputs are the same for each vertex. The commutative diagram is formed by two directed paths for which the same starting and ending points form the full list of common vertices. We refer to the length of the maximal path (the number of arrows) as the rank of the diagram. We will count a directed cycle of length m as a commutative diagram of rank m. We define the** girth indicator gi, gi ≥ 2 **of the directed graph as the minimal rank of its commutative diagram.**

We observe briefly the applications of finite automata related to balanced graphs of high girth in Cryptography. Finally, for each finite commutative ring K with more than two regular elements we consider the explicit construction of algebraic over K family of graphs of high girth and discuss the implementation of the public key algorithm based on finite automata corresponding to members of the family.

I. Introduction

CLASSICAL problems on Turan type problems on studies of the maximal size of simple graphs without prohibited cycles are attractive for mathematicians because they are beautiful and difficult (see [1], [9]). The concept of a family of simple graphs of large girth appears as an important tool to study such problems. Later the applications of these problems in Networking [2], Coding Theory and Cryptography were found (see [15 and further references]).

Section 2 is devoted to the concept of the girth indicator and the family of large girth for digraphs.

In Section 3 we consider the definition of a family of affine algebraic digraphs of large girth over commutative rings. Explicit constructions of such families of graphs can be used for the development of public keys and a key exchange protocol. We discuss the connection of these algorithms with the group theoretical discrete logarithm problem.

The known examples of families of simple algebraic graphs were constructed just in the case of finite fields (see [5]). In section 4 we consider an explicit construction of a family of affine algebraic digraphs of large girth over each finite commutative ring containing at least 3 regular elements. Different properties of this family are investigated in [14], [15], [17] [18] , [20], [7], [8].

Section 5 is devoted to the latest implementation of the public key algorithm based on one of the family described in section 4.

II. On the families of directed graphs of large girth

The missing theoretical definitions on directed graphs the reader can find in [6]. Let Φ be an irreflexive binary relation over the set V, i.e., $\Phi \in V \times V$ and for each v the pair (v, v) is not the element of Φ.

We say that u is the neighbour of v and write $v \to u$ if $(v, u) \in \Phi$. We use the term *balanced binary relation graph* for the graph Γ of irreflexive binary relation ϕ over a finite set V such that for each $v \in V$ the sets $\{x|(x, v) \in \phi\}$ and $\{x|(v, x) \in \phi\}$ have the same cardinality. It is a directed graph without loops and multiple edges. We say that a balanced graph Γ is k-regular if for each vertex $v \in \Gamma$ the cardinality of $\{x|(v, x) \in \phi\}$ is k.

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

We say that the pair of paths $a = x_0 \to x_1 \to \cdots \to x_s = b$, $s \geq 1$ and $a = y_0 \to y_1 \to \cdots \to y_t = b$, $t \geq 1$ 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 $s \geq t$.

We refer to the number $\max(s, t)$ as the rank of $O_{s,t}$. It is ≥ 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, $i = 0, 1, \ldots, s-1$, $s \geq 2$ are distinct vertices.

We will count directed cycles as commutative diagrams.

For the investigation of commutative diagrams we introduce *girth indicator* gi, which is the minimal value for $\max(s, t)$ for parameters s, t of a ommutative diagram $O_{s,t}$, $s + t \geq 3$. The minimum is taken over all pairs of vertices (a, b) in the digraph. Notice that two vertices v and u at distance $<$ gi are connected by the unique path from u to v of length $<$ gi.

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

In case of a 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 $g \geq d+1$. So in the case of the family of graphs with unbounded girth indicator, the girth is also unbounded. We also have $\mathrm{gi} \geq g/2$.

In the case of symmetric irreflexive relations the above mentioned 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 balanced directed regular graphs Γ_i of degree k_i and order v_i such that $\mathrm{gi}(\Gamma_i)$ is $\geq c\log_{k_i} v_i$, where c' is a constant independent of i.

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

The diameter of the strongly connected digraph [6] is the minimal length d of the shortest directed path $a = x_0 \to x_1 \to x_2 \cdots \to x_d$ between two vertices a and b. Recall that a graph is k-regular, if each vertex of G has exactly k outputs. Let F be the infinite family of k_i regular graphs G_i of order v_i and diameter d_i. We say, that F is a family of small world graphs if $d_i \leq C\log_{k_i}(v_i)$, $i = 1, \ldots$ for some constant C independent on i. The definition of small world simple graphs and related explicit constructions the reader can find in [3]. For the studies of small world simple graphs without small cycles see [9], [20] and [33].

III. On the K-theory of affine graphs of high girth and its cryptographical motivations

Let K be a commutative ring. A *directed algebraic graph* ϕ over K consists of two things, such as the *vertex set* Q being a quasiprojective variety over K of nonzero dimension and the *edge set* being a quasiprojective variety ϕ in $Q \times Q$. We assume that ($x\phi y$ means $(x, y) \in \phi$).

The graph ϕ is *balanced* if for each vertex $v \in Q$ the sets $\mathrm{Im}(v) = \{x \,|\, v\phi x\}$ and $\mathrm{Out}(v) = \{x \,|\, x\phi v\}$ are quasiprojective varieties over K of the same dimension.

The graph ϕ is *homogeneous* (or (r, s)-homogeneous) if for each vertex $v \in Q$ the sets $\mathrm{Im}(v) = \{x|v\phi x\}$ and $\mathrm{Out}(v) = \{x|x\phi v\}$ are quasiprojective varieties over F of fixed nonzero dimensions r and s, respectively.

In the case of *balanced homogeneous algebraic graphs* for which $r = s$ we will use the term r-homogeneous graph. Finally, *regular algebraic graph* is a balanced homogeneous algebraic graph over the ring K if each pair of vertices v_1 and v_2 is a pair of isomorphic algebraic varieties.

Let $\mathrm{Reg}(K)$ be the totality of regular elements (or nonzero divisors) of K, i.e., nonzero elements $x \in K$ such that for each nonzero $y \in K$ the product xy is different from 0. We assume that the $\mathrm{Reg}(K)$ contains at least 3 elements. We assume here that K is finite, thus the vertex set and the edge set are finite and we get a usual finite directed graph.

We apply the term *affine graph* for the regular algebraic graph such that its vertex set is an affine variety in Zarisski topology.

Let G be r-regular affine graph with the vertex $V(G)$, such that Out v, $v \in V(G)$ is isomorphic to the variety $R(K)$. Let the variety $E(G)$ be its arrow set (a binary relation in $V(G) \times V(G)$). We use the standard term *perfect algebraic colouring of edges* for the polynomial map ρ from $E(G)$ onto the set $R(K)$ (the set of colours) if for each vertex v different output arrows $e_1 \in \mathrm{Out}(v)$ and $e_2 \in \mathrm{Out}(v)$ have distinct colours $\rho(e_1)$ and $\rho(e_2)$ and the operator $N_\alpha(v)$ of taking the neighbour u of vertex v ($v \to u$) is a polynomial map of the variety $V(G)$ into itself.

We will use the term *rainbow-like colouring* in the case when the perfect algebraic colouring is a bijection. Let $\mathrm{dirg}(G)$ be a directed girth of the graph G, i.e., the minimal length of a directed cycle in the graph. Obviously $\mathrm{gi}(G) \leq \mathrm{dirg}(G)$.

Studies of infinite families of directed affine algebraic digraphs over commutative rings K of large girth with the rainbow-like colouring is a nice and a difficult mathematical problem. Good news is that such families do exist. In the next section we consider the example of such a family for each commutative ring with more than 2 regular elements.

Here, at the end of section, we consider cryptographical motivations for studies of such families.

1) Let G be a finite group and $g \in G$. The discrete logarithm problem for group G is about finding a solution for the equation $g^x = b$ where x is unknown positive number. If the order $|g| = n$ is known we can replace G on a cyclic group C_n. So we may assume that the order of g is sufficiently large to make unfeasible the computation of n. For many finite groups the discrete logarithm problem is NP complete.

Let K be a finite commutative ring and M be an affine variety over K. Then the Cremona group $C(M)$ of all polynomial automorphism of the variety M can be large. For example, if K is a finite prime field F_p and $M = {F_p}^n$ then $C(M)$ is a symmetric group S_{p^n}.

Let us consider the family of affine graphs $G_i(K)$, $i = 1, 2, \ldots$ with the rainbow-like algebraic colouring of edges such that $V(G_i(K)) = V_i(K)$, where K is a commutative ring, and the colour sets are algebraic varieties $R_i(K)$. Let us choose a constant k. The operator $N_\alpha(v)$ of taking the neighbour of a vertex v corresponding to the output arrow of colour α are elements of $C_i = C(V_i(K))$. We can chose a relatively small number k to generate $h = h_i = N_{\alpha_1} N_{\alpha_2} \ldots N_{\alpha_k}$ in each group C_i, $i = 1, 2, \ldots$

Let us assume that the family of graphs $G_i(K)$ is the family of graphs of large girth. It means that the girth indicator $\mathrm{gi}_i = \mathrm{gi}(G_i(K))$ and the parameter $\mathrm{dirg}_i = \mathrm{dirg}(G_i(K))$ are growing with the growth of i. Notice that $|h_i|$ is bounded below by dirg_i/k. So there is j such that for $i \geq j$ the computation of $|h_i|$ is impossible. Finally we can take the base $g = u^{-1}h_j u$ where u is a chosen element of C_j to hide the graph up to conjugation. We may use some package of symbolic computations to express the polynomial map g via the list of polynomials in many unknowns. For example, if $V_j(K)$ is a free module K^n then we can write g in a public mode fashion

$x_1 \to g_1(x_1, x_2, \ldots, x_n)$, $x_2 \to g_2(x_1, x_2, \ldots, x_n)$, $\ldots$, $x_n \to g_n(x_1, x_2, \ldots, x_n)$.

The symbolic map g can be used for Diffie - Hellman *key exchange protocol* (see [3] for the details). Let Alice and Bob be correspondents. Alice computes the symbolic map g and send it to Bob via open channel. So the variety and the map are known for the adversary (Cezar).

Let Alice and Bob choose natural numbers n_A and n_B, respectively.

Bob computes g^{n_B} and sends it to Alice, who computes $(g^{n_B})^{n_A}$, while Alice computes g^{n_A} and sends it to Bob, who is getting $(g^{n_A})^{n_B}$. The common information is $g^{n_A n_B}$ given in "public mode fashion".

Bob can be just a public user (no information on the way in which the map g were cooked) , so he and Cezar are making computations much slower than Alice who has the decomposition $g = u^{-1} N_{\alpha_1} N_{\alpha_2} \ldots N_{\alpha_k} u$.

We may modify slightly the Diffie - Hellman protocol using the action of the group on the variety. Alice chooses a rather short password $\alpha_1, \alpha_2, \ldots, \alpha_k$, computes the public rules for the encryption map g and sends them to Bob via an open channel together with some vertex $v \in V_j(K)$.

Then Alice and Bob choose natural numbers n_A and n_B, respectively.

Bob computes $v_B = g^{n_B}(v)$ and sends it openly to Alice, who computes $(g^{n_A})(v_B)$, while Alice computes $v_A = g^{n_A}(v)$ and sends it to Bob, who is getting $(g^{n_B})(v_A)$.

The common information is the vertex $g^{n_A \times n_B}(v)$.

In both cases Cezar has to solve one of the equations $E^{n_B}(u_A) = z$ or $E^{n_A}(u_B) = w$ for unknowns n_B or n_A, where z and w are known points of the variety.

2) We can construct the *public key* map in the following manner:

The key holder (Alice) chooses the variety $V_j(K)$ and the sequence $\alpha_1, \alpha_2, \ldots, \alpha_t$ of length $t = t(j)$ to determine the encryption map g as above. Let $\dim(V_j(K) = n = n(j)$ and each element of the variety be determined by independent parameters $x_1, x_2, \ldots, x_n$. Alice presents the map in the form of public rules, such as

$x_1 \to f_1(x_1, x_2, \ldots, x_n)$, $x_2 \to f_2(x_1, x_2, \ldots, x_n)$, $\ldots$, $x_n \to f_n(x_1, x_2, \ldots, x_n)$.

We can assume (at least theoretically) that the public rule depending on parameter j is applicable to encryption of potentially infinite text (parameter t is a linear function on j now).

For the computation she may use the Gröbner base technique or alternative methods, special packages for the symbolic computation (popular "Mathematica" or "Maple", package "Galois" for "Java" as well special fast symbolic software). So Alice can use the decomposition of the encryption map into u^{-1}, maps of kind N_α and u to encrypt fast. For the decryption she can use the inverse graph $G_j(K)^{-1}$ for which $VG_j(K)^{-1} = VG_j(K)$ and vertices w_1 and w_2 are connected by an arrow if and only if w_2 and w_1 are connected by an arrow in $G_j(K)$. Let us assume that colours of $w_1 \to w_2$ in $G_j(K)^{-1}$ and $w_2 \to w_1$ in $G_j(K)$ are of the same colour. Let $N'_\alpha(x)$ be the operator of taking the neighbour of vertex x in $G_j(K)^{-1}$ of colour α. Then Alice can decrypt applying consequently $u^{-1}, N'_{\alpha_t}, N'_{\alpha_{t-1}}, \ldots, N_{\alpha_1}$ and u to the ciphertext. So the decryption and the encryption for Alice take the same time. She can use a numerical program to implement her symmetric algorithm.

Bob can encrypt with the public rule but for a decryption he needs to invert the map. Let us consider the case $t_j = kl$, where k is a small number and the sequence $\alpha_1, \alpha_2, \ldots, \alpha_{t_j}$ has the period k and the transformation $h = u^{-1} N_{\alpha_1} N_{\alpha_2} \ldots N_{\alpha_k} u$ is known for Bob in the form of public key mode. In such a case a problem to find the inverse for g is equivalent to a discrete logarithm problem with the base h in related Cremona group of all polynomial bijective transformations.

Of course for further cryptoanalysis we need to study the information on possible divisors of order of the base of related discrete logarithm problem, alternative methods to break the encryption. In the next section the family of digraphs $RE_n(K)$ will be described.

3) We may study security of the private key algorithm used by Alice in the algorithm of the previous paragraph but with a parameter t bounded by the girth indicator of graph $G_j(K)$. In that case different keys produce distinct ciphertexts from the chosen plaintext. In that case we prove that if the adversary has no access to plaintexts then he can break the encryption via the brut-force search via all keys from the key space. The encryption map has no fixed points.

IV. On the family of affine digraph of large girth over commutative rings

E. Moore used term *tactical configuration* of 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, the line set L and the 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 an incidence relation I. We define the following irreflexive binary relation ϕ on the set F:

Let (P, L, I) be the incidence structure corresponding to regular tactical configuration of 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 $p_1 \neq p_2$.

Below we consider the family of graphs $D(k, K)$, where $k > 5$ is a positive integer and K is a commutative ring. Such graphs are disconnected and their connected components were investigated in [17] (for the case when K is a finite field F_q see [5]).

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 points from 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 [16] for the case of general commutative ring K:

$$\begin{aligned}(p) &= (p_{0,1}, p_{1,1}, p_{1,2}, p_{2,1}, p_{2,2}, p'_{2,2}, p_{2,3}, \ldots,\\ &\quad p_{i,i}, p'_{i,i}, p_{i,i+1}, p_{i+1,i}, \ldots),\\ [l] &= [l_{1,0}, l_{1,1}, l_{1,2}, l_{2,1}, l_{2,2}, l'_{2,2}, l_{2,3}, \ldots,\\ &\quad l_{i,i}, l'_{i,i}, l_{i,i+1}, l_{i+1,i}, \ldots].\end{aligned}$$

The elements of P and L can be thought as infinite ordered tuples of elements from K, such that only a 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 coordinates hold:

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

(These four relations are defined for $i \geq 1$, $p'_{1,1} = p_{1,1}$, $l'_{1,1} = l_{1,1}$). This incidence structure (P, L, I) we denote 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 $k \geq 2$ 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 others. The incidence graph corresponding to the structure (P_k, L_k, I_k) is denoted by $D(k, K)$.

For each positive integer $k \geq 2$ 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 projection of each vector from P_k and L_k onto its $k-1$ initial coordinates with respect to the above order.

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

Recall, that we have the arc e' of kind $[l^2, p^2] \to (l^1, p^1)$ 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 $p^1_{1,0} - p^2_{1,0}$. It is easy to see that ρ is a perfect algebraic colouring.

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. We will show that a new graph $RE_n(K)$ $(RE(K))$ with the induced colouring into colours from the alphabet $\mathrm{Reg}(K)$ is vertex transitive. Really, according to [31] graph $D(n, K)$ is an edge transitive. This fact had been established via the description of regular on the edge set subgroup $U(n, K)$ of the automorphisms group $\mathrm{Aut}(G)$. The vertex set for the graph $DE_n(K)$ consists of two copies F_1 and F_2 of the edge set for $D(n, K)$. It means that Group $U(n, K)$ acts regularly on each set F_i, $i = 1, 2$. An explicit description of generators for $U(n, K)$ implicates that this group is a colour preserving group for the graph $DE_n(K)$ with the above colouring.

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., non-zero divisors. Let us delete all arrows with colour, which is a zero divisor. We can show that a new affine graph $RE_n(K)$ $(RE(K))$ with the induced colouring into colours from the alphabet $\mathrm{Reg}(K)$ is vertex transitive (see [18]).

V. On the implementation of the public key algorithm based on $RE(t, K)$

The graphs $CRE_n(K)$ have the best known speed of growth of the girth indicator evaluated in the previous section. It turns out that for the computer implementation of the public key algorithm described in the section 4 the family $RE_n(K)$ of "enveloping" for $CRE_n(K)$ graphs were chosen first. It is also a family of digraphs of large girth but the speed of the growth of girth indicator for the family is less of those for $RE_n(K)$. Graphs $RE_n(K)$ were defined via the family of graphs $D(n, K)$ in the way described in the previous section. So, in some publications the description of the algorithm was done in terms of $D(n, K)$. We introduced here a speed evaluation of the algorithm for its latest implementation.

The set of vertices of the graph $RE_n(K)$ is a union of two copies free module K^{n+1}. So the Cremona group of the variety is the direct product of $C(K^{n+1})$ with itself, expanded by polarity π. In the simplest case of a finite field F_p, where p is a prime number $C(F_p)$ is a symmetric group $S_{p^{n+1}}$. The Cremona group $C(K^{n+1})$ contains the group of all affine invertible transformations, i.e., transformation of kind $\mathrm{x} \to \mathrm{x}A + \mathrm{b}$, where $\mathrm{x} = (x_1, x_2, \ldots, x_{n+1}) \in C(K^{n+1})$, $\mathrm{b} = (b_1, b_2, \ldots, b_{n+1})$ is a chosen vector from $C(K^{n+1})$ and A is a matrix of a linear invertible transformation of K^{n+1}.

Graph $RE_n(K)$ is a bipartite directed graph. We assume that the plaintext K^{n+1} is a point $(p_1, p_2, \ldots, p_{n+1})$. We choose two affine transformations T_1 and T_2 and a linear transformation u will be of kind $p_1 \to p_1 + a_1p_2 + a_3p_3 + \cdots + a_{n+1}$. We slightly modify a general scheme, so Alice computes symbolically of chosen T_1 and T_2, chooses a string $(\beta_1, \beta_2, \ldots, \beta_l)$ of colours for $RE_n(K)$, such that $\beta_i \neq -\beta_{i+1}$ for $i = 1, 2, \ldots, l-1$. She computes $N_l = N_{\beta_1} \times N_{\beta_2} \cdots \times N_{\beta_l}$.

Recall that N_α, $\alpha \in \mathrm{Reg}(K)$ is the operator of taking the neighbour of the vertex v alongside the arrow with the colour α in the graph $RE_n(K)$.

Alice keeps chosen parameters secret and computes the public rule g as the symbolic composition of T_1, N and T_2. The case $uT_2 = {T_1}^{-1}$ is a special form of the general algorithm considered in chapter 4.

In case $K = F_q$, $q = 2^n$ this public key rule has a certain similarity to the Imai-Matsomoto public rule, which is computed as a composition $T_1 E T_2$ of two linear transformations T_1 and T_2 of the vector space $F_{2^n F_{2^s}}$, where F_{2^s} is a special subfield, and E is a special Frobenius automorphism of F_{2^n}. The public rule corresponding to $T_1 E T_2$ is a quadratic polynomial map (see [3] for the detailed description of the algorithm, its cryptoanalisis and generalisations by J. Patarin)

In the case of $RE_n(K)$ the degree of transformation N_l is 3, independently on the choice of length l [21]. So the public rule is a cubical polynomial map of the free module K^{n+1} onto itself. In case of a finite field the algorithm is equivalent to the public rule considered in [19.]

In our computer implementations we used T_1 and T_2 of kind $p_1 \to p_1 + a_1 p_2 + a_3 p_3 + \cdots + a_{n+1} p_{n+1}$.

A. Time evaluation of the private key algorithm for Alice

Alice can use numerical algorithms for the encryption. The decryption ${T_2}^{-1} N_{-\alpha_l} N_{-\alpha_{l-1}} \dots N_{-\alpha_1} {T_1}^{-1}$ takes the same time as the encryption.

In [4] we have implemented a computer application, which uses the family of graphs $RDE(n, K)$ for *private key* cryptography. To achieve high speed, we will use commutative rings $K = Z_{2^k}$, $k \in \{8, 16, 32\}$, with operations $+, \times$ modulo 2^k. The parameter n stands for the length of a plaintext (input data) and the length of a ciphertext. Later on we use a loaded multiplication tables for finite fields and implement cases of finite fields $K = F_{2^k}$, $k \in \{8, 16, 32\}$. We denote 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).

All the tests were run on a computer with parameters:

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

The program is written in Java language. Well known algorithms RC4 which is used for comparison is taken from Java standard library for cryptography purposes - *javax.crypto*.

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

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

TABLE II
TIME OF ENCRYPTION

	$w = 1$ ($\mathbb{Z}_{2^8}$)	$w = 2$ ($\mathbb{Z}_{2^{16}}$)	$w = 4$ ($\mathbb{Z}_{2^{32}}$)
$n = 20$	16	0	0
$n = 40$	265	47	15
$n = 60$	1375	188	15
$n = 80$	3985	578	47
$n = 100$	10078	1360	125

The speed of execution of the above implementation compares well with implementations of other graph based symmetric encryption algorithms [10], [12], [16].

B. On the time evaluation for the public rule

Recall, that we combine a graph transformation N_l with two affine transformation T_1 and T_2. Alice can use $T_1 N_l T_2$ for the construction of the following public map of

$$y = (F_1(x_1, \dots, x_n), \dots, F_n(x_1, \dots, x_n))$$

$F_i(x_1, \dots, x_n)$ are polynomials of n variables written as the sums of monomials of kind $x_{i+1} \dots x_{i_3}$, where $i_1, i_2, i_3 \in 1, 2, \dots, n_1$ with the coefficients from $K = F_q$. As we mention before the polynomial equations $y_i = F_i(x_1, x_2, \dots, x_n)$, which are made public, have the degree 3. Hence the process of an encryption and a decryption can be done in polynomial time $O(n^4)$ (in one y_i, $i = 1, 2 \dots, n$ there are $2(n^3 - 1)$ additions and multiplications). But the cryptoanalyst Cezar, having only a formula for y, has a very hard task to solve the system of n equations of n variables of degree 3. It is solvable in exponential time $O(3^{n^4})$ by the general algorithm based on Gröbner basis method. Anyway studies of specific features of our polynomials could lead to effective cryptoanalysis. This is an open problem for specialists.

We have written a program for generating a public key and for encrypting text using the generated public key. The program is written in C++ and compiled with the Borland bcc 5.5.1 compiler.

We use a matrix in which all diagonal elements equal 1, elements in the first row are non-zero and all other elements are zero as A, identity matrix as B and null vectors as c and d. In such a case the cost of executing affine transformations is linear.

The table **??** presents the time (in milliseconds) of the generation of the public key depending on the number of variables (n) and the password length (p).

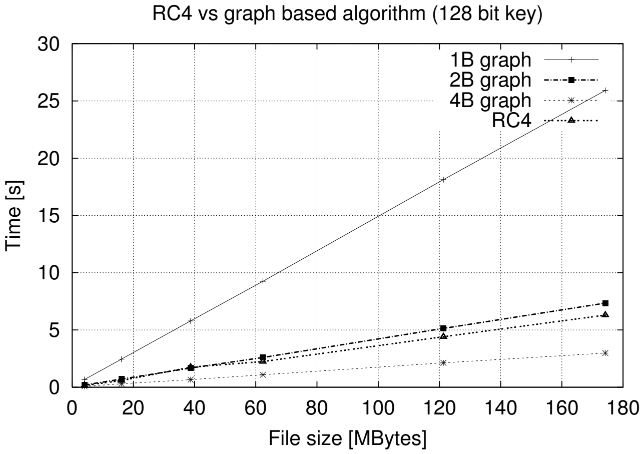
The table **??** presents the time (in milliseconds) of encryption process depending on the number of bytes in plaintext (n) and the number of bytes in a character (w).

REFERENCES

[1] B. Bollobás, *Extremal Graph Theory*, Academic Press, London, 1978.

[2] F. Bien, *Constructions of telephone networks by group representations*, Notices Amer. Mah. Soc., 36 (1989), 5-22.

[3] N. Koblitz, *Algebraic aspects of Cryptography*, in Algorithms and Computations in Mathematics, v. 3, Springer, 1998.



File [MB]	RC4 [s]	G1 [s]	G2 [s]	G4 [s]
4.0	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

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

TABLE I
TIME OF PUBLIC KEY GENERATION

	$p = 10$	$p = 20$	$p = 30$	$p = 40$	$p = 50$	$p = 60$
$n = 10$	15	15	16	32	31	32
$n = 20$	109	250	391	531	687	843
$n = 30$	609	1484	2468	3406	4469	5610
$n = 40$	2219	7391	12828	18219	24484	29625
$n = 50$	5500	17874	34078	49952	66749	82328
$n = 60$	12203	42625	87922	138906	192843	242734
$n = 70$	22734	81453	169250	286188	405500	536641
$n = 80$	46015	165875	350641	619921	911781	1202375
$n = 90$	92125	332641	708859	1262938	1894657	2525360
$n = 100$	159250	587282	1282610	2220610	3505532	4899657

[4] S. Kotorowicz, V. Ustimenko, *On the implementation of cryptoalgorithms based on algebraic graphs over some commutative rings*, Condensed Matter Physics, 2008, vol. 11, No. 2(54), (2008) 347–360.

[5] F. Lazebnik, V. A. Ustimenko and A. J. Woldar, *A New Series of Dense Graphs of High Girth*, Bull (New Series) of AMS, v.32, N1, (1995), 73-79.

[6] R. Ore, *Graph Theory*, Wiley, London, 1971.

[7] T. Shaska, V. Ustimenko, *On the homogeneous algebraic graphs of large girth and their applications*, Linear Algebra and its Applications Article, Volume 430, Issue 7, 1 April 2009, Special Issue in Honor of Thomas J. Laffey.

[8] T. Shaska and V. Ustimenko, *On some applications of graph theory to cryptography and turbocoding*, Special issue of Albanian Journal of Mathematics:Proceedings of the NATO Advanced Studies Institute "New challenges in digital communications", May 2008, University of Vlora, 2008, v.2, issue 3, 249-255.

[9] M. Simonovits *Extremal Graph Theory*, Selected Topics in Graph Theory 2 (L.W. Beineke and R.J. Wilson, eds), Academic Press, London, 1983, 161-200.

[10] A. Touzene, V. Ustimenko, *Private and Public Key Systems Using Graphs of High Girth*, in Roland E. Chen (editor), Cryptography Research Perspective, Nova Science Publishers, April, 2009.

[11] V. Ustimenko, *Graphs with Special Arcs and Cryptography*, Acta Applicandae Mathematicae, 2002, vol. 74, N2, 117-153.

[12] V. Ustimenko, *CRYPTIM: Graphs as tools for symmetric encryption*, In Lecture Notes in Comput. Sci., 2227, Springer, New York, 2001.

[13] V. A. Ustimenko, *Linguistic Dynamical Systems, Graphs of Large Girth and Cryptography*, Journal of Mathematical Sciences, Springer, vol.140, N3 (2007) pp. 412-434.

[14] V. Ustimenko, *On the extremal regular directed graphs without commutative diagrams and their applications in Coding Theory and Cryptography*, Special Issue of Albanian J. Math. on Algebra and Computational Algebraic Geometry, Vol.1, No 4 (2007).

[15] V. A. Ustimenko, *On the extremal graph theory for directed graphs and its cryptographical applications*, in T. Shaska, W. C. Huffman, D. Joener and V. Ustimenko (editors), Advances in Coding Theory and Cryptography. Series on Coding Theory and Cryptology, World Scientific, vol. 3, (2007).

[16] V. Ustimenko and J. Kotorowicz, *On the Properties of Stream Ciphers Based on Extremal Directed Graphs*, in Roland E. Chen (editor), Cryptography Research Perspective, Nova Science Publishers, April 2009.

[17] V. Ustimenko, *Algebraic groups and small world graphs of high girth*, Albanian J. Math., Vol. 3, No 1 (2009), 25-33.

[18] V. A. Ustimenko, *On the cryptographical properties of extremal algebraic graphs*, in Algebraic Aspects of Digital Communications, IOS Press (Lectures of Advanced NATO Institute, NATO Science for Peace and Security Series - D: Information and Communication Security, Volume 24, July 2009, 296 pp.

[19] V. Ustimenko, *Maximality of affine group and hidden graph cryptsystems*, Journal of Algebra and Discrete Mathematics, October, 2004, v.10, pp. 51-65.

[20] V. Ustimenko,*On the extremal regular directed graphs without commutative diagrams and their applications in Coding Theory and Cryptography*, 2007, Albanian Journal of Mathematics, 2007 (Special Issue on Algebra and Computational Algebraic Geometry), Vol 1, N4 (2007).

[21] A. Wróblewska, *On some properties of graph based public key*, Albanian J. Math., vol. 2, No 3 (2008), Special Issue "New Challenges of Digital Communications", Proc. of NATO Advanced Studies Institute, Vlora, 2008, pp. 229-234.