ALBANIAN JOURNAL OF
MATHEMATICS
Volume 5, Number 3, Pages 139–149
ISSN: 1930-1235; (**2011**)

ON THE IMPLEMENTATION OF PUBLIC KEY ALGORITHM BASED ON GRAPHS AND THEIR SYMMETRIES

Michał Klisowski
University of Maria Curiwe Sklodowska, Poland
Email: mklisow@hektor.umcs.lublin.pl

Vasyl Ustimenko
University of Maria Curiwe Sklodowska, Poland
Email: vasyl@hektor.umcs.lublin.pl

Abstract. The paper devoted to implementation of the public key algorithm based on directed algebraic graphs over finite commutative ring K and their symmetries. First we expand the key space K^n of graph based encryption algorithm in such way that arbitrary chosen plaintext can be converted to arbitrary chosen ciphertext. Second, we conjugate chosen encryption map, which is a composition of several "elementary" cubical polynomial automorphisms of a free module K^n with special invertible affine transformation of K^n . Finally we compute symbolically corresponding cubic public map g of K^n onto K^n. We evaluate time for the generation of g, time of execution of public map, number of monomial expression in the list of corresponding public rules.

1. Introduction

Cloud computing systems open a new perspective in various aspects of computing.

Some security issues raised by cloud computing are motivated by virtualization. Dynamic scalability or "elasticity" will help generalize high-performance computing and very large data sets in applications. But the real gains in performance depend heavily on the predictability of physical and virtualized resources. It means that the balancing of performance against security and the adaptation of HPC or VLDB techniques to cloud computing are important issues and will have long-lasting scientific content.

The direction of Key Dependent Message (KDM) secure encryption in Cryptography can bring an appropriate security tools for Cloud Computing.

In publications [4] were proposed classes of stream ciphers and public key algorithms based on explicit construction of families of algebraic graphs of large girth.

It was shown that for each finite commutative ring K we can create a cubical polynomial map f of K^n onto K^n depending on string of regular elements (non zero divisors $(\alpha_1\alpha_2, \ldots, \alpha_t)$ password). If $t \leq (n+5)/2$ then different strings produce different ciphertext. One can use such a map as a stream cipher. It is possible to combine f with two invertible sparse affine transformations τ_1 and τ_2 and use the composition $g = \tau_1 f \tau_2$ as a public rule. Public user is not able to decrypt without knowledge of τ_1, τ_2 and string $(\alpha_1\alpha_2, \ldots, \alpha_t)$.

One can set τ_2 as the inverse of τ_1 and use the "symbolic" generator g and related cyclic groupfor the Diffie -Hellman key exchange protocol. We can prove that the order of g is growing with the grows of parameter n

This publication is devoted to the implementation of generalisation of the above algorithm. We consider linear transformations T_{a} depending on the string $\mathrm{a} = (\beta_1, \beta_2, \ldots, \beta_d)$, where$d = [n/4]$ and use fT_{a} instead of f.

The constraction of transformation f use graphs $D(n, K)$ (graphs of large girth for $K = F_q$, which was very useful for creation of good LDPS codes in Coding Theory. The transformation T_{a} is a special automorphism of graph $D(n, K)$.

In fact the key space of all passwords $g = fT_{\mathrm{a}}$ has the following property in case of char k-for each pair plaintext p - ciphertext c there is a transform g sending p to c. So we hope that usage of families of large girth and their autompprphism may lead to good public keys.

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 [2], [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 [1], Coding Theory and Cryptography were found (see [11] 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 [12], [11], [13], [14] , [8], [7].

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

2. 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 $a = x_0 \to x_1 \to \dots x_s = b$ of length s, where x_i, $i = 0, 1, \dots s$ 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 $O_s = O_{s,0}$: $v_0 \to v_1 \to \dots v_{s-1} \to v_0$, where v_i, $i = 0, 1, \dots, s-1$, $s \geq 2$ 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 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 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 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 gi(Γ_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 $d_i \leq C\log_{k_i}(v_i)$, $i = 1, \dots$ 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], [12].

3. 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 $h = h_i = N_{\alpha_1} N_{\alpha_2} \dots N_{\alpha_k}$ in each group C_i, $i = 1, 2, \dots$

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

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

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 $g = u^{-1} N_{\alpha_1} N_{\alpha_2} \dots N_{\alpha_k} u$.

We may modify slightly the Diffie - Hellman protocol using the action of the group on the variety. Alice chooses a rather short password $\alpha_1, \alpha_2, \dots, \alpha_k$, 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 $\alpha_1, \alpha_2, \dots, \alpha_t$ 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 $x_1, x_2, \dots, x_n$. Alice presents the map in the form of public rules, such as

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

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 $u^{-1}, N'_{\alpha_t}, N'_{\alpha_{t-1}}, \dots, N_{\alpha_1}$ 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 $\alpha_1, \alpha_2, \dots, \alpha_{t_j}$ has the period k and the transformation $h = u^{-1}N_{\alpha_1}N_{\alpha_2}\dots N_{\alpha_k}u$ 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.

4. 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 [13] (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 [15] for the case of general commutative ring K:

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

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 co-ordinates 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. The transformation $t'_{m,m}(x)$ acts on vertices of $D(K)$ by the following rules.

(a) $l'_{m,m} \to l'_{m,m} + x$, $p'_{m,m} \to p'_{m,m} + x$.

(b) $l_{m+1,m} \to l_{m+1,m} + l_{1,0}x$.

(c) $l_{m+1,m+1} \to l_{m+1,m+1} + l_{1,1}x$, $p_{m+1,m+1} \to l_{m+1,m+1} + p_{1,1}x$

(d) $l_{m+r,m+r} \to l_{m+r,m+r} + l'_{r,r}x$, $p_{m+r,m+r} \to p_{m+r,m+r} + p'_{r,r}x$, $r \geq 2$.

(e) $l_{m+r+1,m+r} \to l_{m+r+1,m+r} + l_{r+1,r}x$, $p_{m+r+1,m+r} \to p_{m+r+1,m+r} + p_{r+1,r}x$, $r \geq 2$.

(f) All other components are unchanged.

We define the transformation T_{a}, where $\mathrm{a} = (\beta_{22}, \beta_{33}, \dots)$ as a product of all transformations $t'_{i,i}(\beta_{ii})$

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 [9] 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 [14]).

Notice, that each T_{a} acts naturally on the flags, it is an automorphism of $RE_n(K)$.

5. 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 $p_1 \to p_1 + a_1 p_2 + a_3 p_3 + \cdots + a_{n+1}$. 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 chooses additionally string a.

Alice keeps chosen parameters secret and computes the public rule g as the symbolic composition of T_1, N, T_a and T_2.

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_1ET_2 of two linear transformations T_1 and T_2 of the vector space $F_{2^nF_{2^s}}$, 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_1ET_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 [16]. 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 [10].

5.1. **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 and transformation T_a. Alice can use $T_1N_lT_aT_2$ 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 gcc compiler (version 4.4.1).

We have implemented three cases:

- T_1 and T_2 are identities,
- T_1 and T_2 are of kind $x_1 \to x_1 + a_2x_2 + a_3x_3 + \dots + a_{n+1}x_{n+1}$ (linear time of computing T_1 and T_2),
- $T_1 = A_1x + b_1$, $T_2 = A_2x + b_2$; matrices A_1, A_2 and vectors b_1, b_2 has mostly nonzero elements.

The table 1 applies to the second case. It presents the time (in milliseconds) of the generation of the public key depending on the number of variables (n) and the password length (p). It also presents the time of computing the transformation T_a.

The time of encryption process depends linearly on the number of monomials (the number of nonzero coefficients) in cubic polynomials F_1, $F_2 \dots F_n$ in the public map $y = (F_1(x_1, \dots, x_n), \dots, F_n(x_1, \dots, x_n))$. For $n = 120$ and $p = 60$ this number is about 8500 in the first case, about 780000 in the second case and about 2600000 in the third case.

TABLE 1. Time of public key generation ($K = F_{2^{32}}$)

	$p = 20$		$p = 40$		$p = 60$	
		T_a		T_a		T_a
$n = 10$	16	0	16	0	31	0
$n = 20$	141	16	280	0	437	0
$n = 30$	562	0	1217	0	1888	16
$n = 40$	1513	16	3464	16	5678	16
$n = 50$	3261	63	8346	62	13089	62
$n = 60$	6271	125	16239	140	26426	156
$n = 70$	11139	328	29032	328	47440	312
$n = 80$	17301	531	47315	546	79279	561
$n = 90$	26582	1435	72415	1560	122866	1513
$n = 100$	38173	2418	104053	2418	180790	2418
$n = 110$	53149	3557	144987	3634	251380	3572
$n = 120$	70169	4867	189479	3151	338258	3308

Applying the transformation T_a has the biggest impact on the time of encryption in the first case — about 16% for $n = 120$ and $p = 60$. In the second case it is about 3.5% and in the third case it has no impact at all. The tables 2 and 3 apply to the second case. They present the number of monomials in a public map depending on n and p. The table 2 shows the number of monomials in a public map without transformation T_a and the table 3 — the number of monomials in a public map with T_a.

TABLE 2. Number (percentage) of nonzero coefficients (T_1NT_2)

	$p = 20$	$p = 40$	$p = 60$
$n = 10$	435 (15.21%)	435 (15.21%)	435 (15.21%)
$n = 20$	3327 (9.39%)	3327 (9.39%)	3327 (9.39%)
$n = 30$	11795 (7.21%)	11795 (7.21%)	11795 (7.21%)
$n = 40$	27426 (5.56%)	27427 (5.56%)	27427 (5.56%)
$n = 50$	49995 (4.27%)	54855 (4.68%)	54855 (4.68%)
$n = 60$	77245 (3.24%)	93552 (3.93%)	93552 (3.93%)
$n = 70$	110395 (2.54%)	150865 (3.47%)	150865 (3.47%)
$n = 80$	149445 (2.03%)	222951 (3.03%)	222952 (3.03%)
$n = 90$	194395 (1.66%)	307015 (2.63%)	321075 (2.75%)
$n = 100$	245245 (1.39%)	398140 (2.25%)	436877 (2.47%)
$n = 110$	301995 (1.17%)	501165 (1.95%)	586735 (2.28%)
$n = 120$	364645 (1.00%)	616090 (1.70%)	756576 (2.08%)

REFERENCES

[1] F. Bien, *Constructions of telephone networks by group representations*, Notices Amer. Math. Soc. **3** (1989), 5–22.
[2] B. Bollobás, *Extremal graph theory*, Academic Press, London, 1978.
[3] N. Koblitz, *Algebraic aspects of cryptography*, Algorithms and Computation in Mathematics, vol. 3, Springer, 1998.

TABLE 3. Number (percentage) of nonzero coefficients ($T_1NT_aT_2$)

	$p = 20$	$p = 40$	$p = 60$
$n = 10$	435 (15.21%)	435 (15.21%)	435 (15.21%)
$n = 20$	3367 (9.51%)	3367 (9.51%)	3367 (9.51%)
$n = 30$	12070 (7.37%)	12070 (7.37%)	12070 (7.37%)
$n = 40$	28126 (5.70%)	28127 (5.70%)	28127 (5.70%)
$n = 50$	52009 (4.44%)	56505 (4.82%)	56505 (4.82%)
$n = 60$	81653 (3.43%)	96412 (4.05%)	96412 (4.05%)
$n = 70$	118894 (2.73%)	155865 (3.58%)	155865 (3.58%)
$n = 80$	158948 (2.16%)	230346 (3.13%)	230347 (3.13%)
$n = 90$	221795 (1.90%)	318860 (2.73%)	332275 (2.85%)
$n = 100$	287881 (1.63%)	416661 (2.36%)	452057 (2.56%)
$n = 110$	361270 (1.40%)	529460 (2.06%)	607860 (2.36%)
$n = 120$	448691 (1.24%)	646858 (1.78%)	783666 (2.16%)

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

[5] F. Lazebnik, V. A. Ustimenko, and A. J. Woldar, *A new series of dense graphs of high girth*, Bull. Amer. Math. Soc. (N.S.) **32** (1995), no. 1, 73–79.

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

[7] T. Shaska and V. Ustimenko, *On some applications of graph theory to cryptography and turbocoding*, Albanian J. Math. **2** (2008), no. 3, 249–255, Proceedings of the NATO Advanced Studies Institute: "New challenges in digital communications".

[8] ———, *On the homogeneous algebraic graphs of large girth and their applications*, Linear Algebra Appl. **430** (2009), no. 7, 1826–1837, Special Issue in Honor of Thomas J. Laffey.

[9] M. Simonovits, *Extremal graph theory*, Selected Topics in Graph Theory 2 (L. W. Beineke and R. J. Wilson, eds.), no. 2, Academic Press, London, 1983, pp. 161–200.

[10] V. Ustimenko, *Maximality of affine group and hidden graph cryptosystems*, J. Algebra Discrete Math. **10** (2004), 51–65.

[11] ———, *On the extremal graph theory for directed graphs and its cryptographical applications*, Advances in Coding Theory and Cryptography (T. Shaska, D. W. C. Huffman, Joener, and V. Ustimenko, eds.), Series on Coding Theory and Cryptology, vol. 3, World Scientific, 2007, pp. 181–199.

[12] ———, *On the extremal regular directed graphs without commutative diagrams and their applications in coding theory and cryptography*, Albanian J. Math. **1** (2007), no. 4, Special issue on algebra and computational algebraic geometry.

[13] ———, *Algebraic groups and small world graphs of high girth*, Albanian J. Math. **3** (2009), no. 1, 25–33.

[14] ———, *On the cryptographical properties of extremal algebraic graphs*, Algebraic Aspects of Digital Communications (Tanush Shaska and Engjell Hasimaj, eds.), NATO Science for Peace and Security Series - D: Information and Communication Security, vol. 24, IOS Press, July 2009, pp. 256–281.

[15] V. Ustimenko and J. Kotorowicz, *On the properties of stream ciphers based on extremal directed graphs*, Cryptography Research Perspective (Roland E. Chen, ed.), Nova Science Publishers, April 2009, pp. 125–141.

[16] A. Wróblewska, *On some applications of graph based public key*, Albanian J. Math. **2** (2008), no. 3, 229–234, Proceedings of the NATO Advanced Studies Institute: "New challenges in digital communications".