Algebra and Discrete Mathematics
Number 1. **(2005).** pp. 133 – 150
© Journal "Algebra and Discrete Mathematics"

RESEARCH ARTICLE

Maximality of affine group, and hidden graph cryptosystems

Vasiliy A. Ustimenko

Communicated by V. I. Sushchansky

Dedicated to Yu.A. Drozd on the occasion of his 60th birthday

Abstract. We describe a new algebraic-combinatorial method of public key encryption with a certain similarity to the well known Imai-Matsumoto. We use the general idea to treat vertices of a linguistic graph (see [21] and further references) as messages and use the iterative process to walk on such graph as encryption process. To hide such encryption (graph and walk on it) we will use two affine transformation. Like in Imai - Matsumoto encryption the public rule is just a direct polynomial map from the plaintext to the ciphertext.

The knowledge about graph and chosen walk on them (the key) allow to decrypt a ciphertext fast. We hope that the system is secure even in the case when the graph is Public but the walk is hidden. In case of "public" graph we can use same encryption as private key algorithm with the resistance to attacks when adversary knows several pairs:(plaintext, ciphertext).

We shall discuss the general idea of combining affine transformations and chosen polynomial map of deg ≥ 2 in case of prime field F_p. As it follows from the maximality of affine group each bijection on ${F_p}^n$ can be obtained by such combining.

1. Introduction

We are getting lots of information through e-mail, e-commerce (business conducted over the Internet), cellular phones, or satellite and cable

Key words and phrases: *Data and communication security, e-commerce, Public Key Cryptography, Private Key Encryption.*

TV. The need for secure communications is more profound than ever, for instance, the emergence of virtual organisations such as e-learning and e-business in general shall rely on a secure way for online transactions. In order for a data to be secured for storage or transmission, it must be transformed in such a manner that it would be difficult for an unauthorised person to be able to discover its true meaning. This is the purpose of cryptographic techniques. Development of new theoretical tools for data protection and computational testing of their software implementations is active area of research.

The idea to use arcs on graphs for encryption (section 3) has been considered in [16],[17]. In this paper we shall combine "graphical encryption" with two affine transformation (section 5) to get a public and private key algorithms.

First (section 2), we discuss general idea of combining of different affine transformations with the chosen polynomial map. In section 4 we describe Imai-Matsumoto scheme following to [4], so the reader can compare it with the graphical public key algorithm in section 5. To describe public key algorithms we shall use traditional for the Cryptography characters: Alice (the holder of the key), Bob (the public user) and Catherine (the cryptoanalyst). In section 6 we consider some heuristic arguments on the complexity of algorithms in case of linguistic graphs of high girth, infinite family of such graphs. The toy example of proposed public key encryption based the reader can find at the end of the article.

Papers [18]-[20], [23], [24] devoted to the implementations of the algorithm using walks on graphs.

2. Maximality of affine group and Cryptography

Let $(F_q)^n$ be a vector space over the finite field F_q, where q is the prime power.

As it is usual in cryptography, we can apply a term *plaintext* to a string of characters $\mathrm{x} = (x_1, x_2, \ldots, x_n)$ over the alphabet F_q. When working with matrices, we shall consider vectors to be column vectors (although in the text we shall continue use them as rows). We can consider x as a message containing a certain information. If π is some bijective transformation of $(F_q)^n$, then $\pi(\mathrm{x})$ is an encrypted message or a *ciphertext*.

The natural choice for π is a combination of some affine transformations $a_i = A_i\mathrm{x} + b_i$, $i = 1, \ldots, k$, where A_i is a square matrix and $b_i \in (F_q)^n$, with some nonlinear transformation T of the vector space $(F_q)^n$.

Let us consider the case of F_p, where p is a prime number. Affine transformations $\mathrm{x} \to A\mathrm{x} + b$, where A be an invertible matrix and $b \in$

$(F_q)^n$ form an affine group $AGL_n(F_p)$ of order $p^n(p^n-1)(p^n-p)\dots(p^n-p^{n-1}$. This is a subgroup of the symmetric group S_{p^n} of order $(p^n)!$.

The following fact had been proven in [14].

Theorem 2.1. Let G a proper subgroup of S_{p^n} containing $AGL_n(F_p)$. Then G coincides with $AGL_n(p)$ or S_{p^n}.

Let us choose the nonlinear transformation T. The following statement follows directly from the theorem

Corollary 2.2. Let T be a chosen nonaffine transformation of the vector space $V = {F_p}^n$. Then each bijective transformation T of the vector space $V = (F_p)^n$ can be presented as a product of "quantum" transformations $Q(\alpha_1, \alpha_2) = alpha_1 T\alpha_2$ where α_1 and α_2 are appropriate affine transformations of V.

We recall the following well-known algebraic fact:

Theorem 2.3. Each transformation T of the vector space $V = (F_p)^n$ can be treated as a polynomial map $P : \mathrm{x} \to \mathrm{y}$, where $\mathrm{x} = (x_1, \dots, x_n)$, $\mathrm{y} = (y_1, \dots, y_n)$, $y_i = P_i(x_1, \dots, x_n)$, $i = 1, \dots, n$ for some polynomial expressions P_i in variables x_i over the finite field F_p.

The following "public key" strategy can be derived naturally from the statements above:

(A) Choose polynomial transformation P, which you can invert fast (for polynomial number of steps $f(n)$), where deg$f(n)$ is "small")

(B) select the family Ω of affine transformations $\alpha_i, i = 1, \dots, m$ and quantum maps $Q_j = \beta_j P\gamma_j$, $j = 1, \dots, l$, where $\beta_j, \gamma_j \in \Omega$

(C) compute the polynomial map $Q = Q_1 Q_2 \dots Q_l$ (composition of Q_i),i. e get the formula $y = Q(x) = (P_1(x_1, \dots x_n), \dots, P_n(x_1, \dots, x_n))$, where polynomials P_i written in canonical form.

(D) keep transformations Q_i, α_j and the decomposition of Q into quantum maps Q_i secret and give the list L of public equations P_i to you correspondent B.

Now, you correspondent can encrypt his/her messages to you by applying Q to the plaintext, but the problem of decryption , i.e. computation of inverse map Q^{-1} can reach any level of complexity: you may obtain any permutation from the symmetric group of S_{p^n} as your expression Q.

For you, the problem, of decryption can be feasible if length l is "reasonably moderate". You can invert each Q_i and apply them to the ciphertext in the reverse order with respect to the known decomposition of Q in Q_i.

Of course, we have no illusion to solve mathematically the great problem of cryptography on the existence of asymmetric function: corollary from the theorem 1 does not contain any restrictions on the number l of quantum transformations.

But, it is reasonable to assume that even in case of polynomial length l we are able to produce practically secure public keys. In fact, well known Imai-Matsumoto encryption scheme and its modifications by J. Patarin are realisation of A-D in case $l = 1$. They are "quantum transformations". Double round of these algorithms corresponds to the case of $l = 2$. Of course, for any encryption (RSA, in particular) exists a presentation in the form A-D, but we have just theorem of existence without effective algorithms to construct a decomposition of given permutation into composition of quantum transformations.

Remark 1. Substitution the field $GF(q)$, where $q = p^j$, p is a prime number, instead of F_p in the scheme (A)-(D) does not led to more general scheme, because of vector space $(F_q)^d$ over the ground field F_q, is a vector space of dimension jd over F_p, but such a substitution can be useful in practical applications. We can consider also a K^j, where K is a commutative field, instead of vector space ${F_p}^j$.

Remark 2. Size of the family Ω of step B can be bounded by polynomial expression in variable n, we may think that Ω consist of some elementary transvections $t_{i,j}(1)$, $i \neq j$ and diagonal matrices for which exactly one entry equals to fixed generator of of multiplicative group of F_p and other entries equal 1, regular translations $\mathrm{x} \to \mathrm{x} + \mathrm{e}_i$, where e_i is addition of 1 to x_i. One can consider even smaller set of generators of Affine group.

To compare public polynomial maps of kind $P = (P_1, \dots, P_n)$ 'in glance" one may look on the degrees of polynomials $P_i(x_1, x_2, x_n)$, or Grobner basis of the ideal generated by P_i. Like in case of RSA, heuristic arguments are useful to convince public that problem of inverting P is hard.

3. Parallelotopic and Linguistic graphs as tools for the encryption

In this subsection we will consider the *parallelotopic graphs* for which arcs can be identified naturally and effectively with words in a certain alphabet M without marking of edges. We can just paint the vertices of our graph for this purpose.

We say [17], [21] that $\Gamma = (\Gamma, M, \pi)$ is a *parallelotopic graph* over a finite set M if we have a surjective function $\pi : V(\Gamma) \to M$ such that for every pair (v, m), $v \in V(\Gamma)$, $m \in M$, there is a unique neighbour u of v satisfying $\pi(u) = m$.

given vertex v, and any colour m, there exists exactly one neighbour u of v of colour m.

Let Γ be a parallelotopic graph. We shall treat its vertices as plaintexts. So the set of vertices $V(\Gamma)$ is the plainspace and cipherspace. Let $N(t, v)$ be the operator taking the neighbour u with colour t of a vertex v of a parallelotopic graph Γ. If $(t_1, t_2, \cdots, t_n)$, $t_i \in M$ is a tuple such that $t_i \neq t_{i+2}$, then $(v, v_1 = N(t_1, v), v_2 = N(t_2, v_1), \cdots, v_n = N(t_n, v_{n-1}))$ is the arc of the graph Γ which we can consider as an *encoding arc* for any chosen vertex v.

Let us refer to this tuple $\rho = (t_1, \cdots, t_n)$ over M as an *encoding tuple.* It is clear that $\rho^{-1} = (t_{n-1}, \cdots, t_1, c(v))$ is the "decoding tuple", because it corresponds to the decoding arc.

We can modify this definition in case of bipartite graphs.

Let Γ be a bipartite graph with partition sets P_i, $i = 1, 2$. Let M be a disjoint union of finite sets M_1 and M_2. We say that Γ is a bipartite parallelotopic graph over (M_1, M_2) if there exists a function $\pi : V(\Gamma) \to M$ such that if $p \in P_i$, then $\pi(p) \in M_i$ and for every pair (p, j), $p \in P_i$, $j \in M_i$, there is a unique neighbour u with given $\pi(u) = j$.

It is clear that the bipartite parallelotopic graph Γ is a $(|M_1|, |M_2|)$ - biregular graph.

We refer also to the function π in the definition of bipartite parallelotopic graph also as a labelling. We will often omit the term "bipartite ", because all our graphs are bipartite.

Let M^t be the Cartesian product of t copies of the set M.

We say that the graph Γ is a linguistic graph over the set M with parameters m, k, r, s if

Γ is a bipartite parallelotopic graph over (V_1, V_2), $M_1 = M^r$, $M_2 = M^s$ with the set of points $I = M^m$ (inputs) and set of lines $O = M^k$ (outputs). (i.e. M^m and M^k are the partition sets of Γ). It is clear that $m + r = k + s$.

We use the term linguistic coding scheme for a pair (Γ, n), where Γ is linguistic graph and $n < g$ is the length of encoding sequences.

We choose a bipartite graph in the definition above because regular trees are infinite bipartite graphs and many biregular finite graphs of high girth can be obtained as their quotients (homomorphic images).

For linguistic graphs our messages and coding tools are words over the alphabet M and we can use the usual matching between real information and vertices of our graph.

We use the term linguistic graph over $GF(q)$ when we have a linguistic graph with alphabet $M = GF(q)$ and the set of neighbours of any vertex v is an algebraic manifold over $GF(q)$, i.e. is the totality of solutions of a certain system of polynomial equations. Here,

our messages (plaintexts or ciphertexts) and enryption tools (passwords) are tuples over $GF(q)$.

The following construction proves the existence of linguistic graphs, shows that there are "many" of them on $2q^n$ vertices, where q is arbitrary prime power and n is arbitrary positive integer.

Let P and L be two n-dimensional vector spaces over $GF(q)$. 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 chose two fixed bases and write:

$$(p) = (p_1, \dots, p_n)$$

$$[l] = [l_1, \dots, l_n]$$

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

$$a_2 l_2 - b_2 p_2 = P_2(l_1, p_1)$$

$$\dots$$

$$a_i l_i - b_i p_i = P_i(l_1, \dots, l_{i-1}, p_1, \dots, l_{i-1}) \qquad (1)$$

$$\dots$$

$$a_n l_n - b_n p_n = P_n(l_1, \dots l_{n-1}, p_1, \dots, p_{n-1})$$

where P_i, $i = 2, \dots, n$ can be any polynomial expressions in variables $l_1, \dots, l_{i-1}, p_1, \dots, p_{i-1}$ over F_q, a_i, b_i can be any nonzero elements from F_q.

It is easy to see that the above graph is a linguistic graph such that p_1 and l_1 are the "colours" of (p) and $[l]$, respectively, and $r = s = 1$. If $q = 2$, then the bipartite graph defined by equations (1) are disjoint union of cycles.

Let us call the graphs defined by equations of kind (1) *linguistic graphs of triangular type over* F_q.

4. The Imai - Matsumoto scheme

Let K be an extension of degree n of the finite field F_q, where q is a power of 2, and let $\beta_1, \beta_2, \ldots, \beta_n$ be a basis of K as an F_q-vector space.

Alice will be using the Imai- Matsymoto system in K. She regards each element of K as an n-tuple over q.

Alice may choose to keep her basis secret in which case we can not assume that a cryptoanalyst (whom we shall name "Catherine") knows what basis she is using.

Both plaintext message units and ciphertext message units will be n-tples over F_q. We will use the vector notation $\mathrm{x} = (x_1, x_2, \ldots, x_n) \in {F_q}^n$ for plaintext and $\mathrm{y} = (y_1, y_2, \ldots, y_n) \in {F_q}^n$ for ciphertext. When working with matrices, we shall consider vectors to be column vectors (although in the text we shall continue writing them as rows).

In transforming paintext into ciphertext, Alice will work two intermidiate vectors, denoted $\mathrm{u} = (u_1, \ldots, u_n) \in {F_q}^n$ and $\mathrm{v} = (v_1, \ldots, v_n) \in {F_q}^n$. Given a vector in ${F_q}^n$, we shall use boldface to denote the corresponding element of K with respect to the basis β_j.

Next, Alice chooses an exponent h, $0 < h < q^n$, that is of the form

$h = q^\alpha + 1$

and satisfies the condition g.c.d.$(h, q^n - 1) = 1$. (Recall that q was choosen to be a power of 2, if q were odd, then g.c.d.$(h, q^n - 1)$ is at least 2.)

The condition g.c.d.$(h, q^n - 1) = 1$ is equivalent to requiring that the map $\mathrm{u} \to \mathrm{u}^h$ on K is one to one, its inverse is the map $\mathrm{u} \to \mathrm{u}^{h'}$, where h' is the multiplicative inverse of h modulo $q^n - 1$.

Alice may choose o keep h secret. However, since there are relatively few possible values for h, she must asume that Catherine will be prepared to run through all possibilities for h. That is, even if she keeps h secret, the security of her system must be elsewhere.

In addition, Alice chooses two secret affine transformations, i. e., two invertible $n \times n$ matrices $A = (a_{ij})$, $1 \le i, j \le n$ and $B = (b_{ij})$, $1 \le i, j \le n$ with entries in F_q, and two constant vectors $\mathrm{c} = (c_1, \ldots, c_n)$ and $\mathrm{d} = (d_1, \ldots, d_n)$.

The purpose of the two transformations is to "hide the monomial map" $u \to u^h$ - hence the name "hidden monomial cryptosystem".

We now describe how Alice gets her public rule for going from plaintext $\mathrm{x} \in {F_q}^n$ to ciphertext $\mathrm{y} \in {F_q}^n$.

First, she sets

$\mathrm{u} = A\mathrm{x} + \mathrm{c}$.

Next, she would like to have $v \in K$ simply equal to the h-th power of $\mathrm{u} \in K$ and then set

y $= B^{-1}$(v − d)
that is v $= B$y + d.

However, her public encryption rule will go right from x to y, and will not directly involve exponentiation at all.

In order to get formulas going from x directly to y Alice notices that since $v = u^h$ and $h = q^\theta + 1$, she has
$v = u^{q^\theta} u$.

Recall that for any $k = 1, 2, ..., n$ the operation of raising to the q^k-th power in K is an F_q-linear transformation. Using linear algebra, she can get n-equations that express each y as a polynomial of total degree 2 in the $x_1, \dots, x_n$.

Alice makes these n equations public. If Bob wants to send her a plaintext message x, he substitutes the x_i in these equations and finds y_i. On the other hand, Catherine, who knows only the ciphertext (and the public key), must solve a nonlinear system for the unknowns x_i.

When Alice receives the iphertext y, she uses her knoledge of A, B, c and d and h to recover x, without having to solve the publicly known equations for the x_i. Namely, let h′ be the multiplicative inverse of h modulo $q^n - 1$, so that the map $u = v^{\mathrm{h}'}$ inverts the map $v = u^{\mathrm{h}}$ on K. Alice first computes v $= B$y $+ d$, then raises v$= \sum v_i\beta_i \in K$ to the h′-th power (i.e., sets $u = v^{\mathrm{h}'}$, and finaly compute x $= A^{-1}$(u − c.

The following summarises Alice's decryption:
$(y_1, \dots, y_n) \to$ y $= B$y $+ d \to$ u $=$ v$^{\mathrm{h}'} \to$ x $= A^{-1}$(u − c)

Remark. The cryptosystem described above is a simplified version of the one proposed in the original paper [6]. For details about breaking the original Imai-Matsumoto system see [15].

5. Hidden linguistic graphs system, hidden walks systems

Let F_q, $q > 2$ be the finite field, where q is a prime power.

Alice shall be using the hidden graph scheme basd on the family of linguistic graphs $L_n(q)$ with the operators $N_c(v)$ to take the neighbour u of vertex v such that c is the colour of u.

Let $I_1(x_1, \dots, x_n), \dots, I_k(x_1, \dots, x_n)$ are some invariants of graph $L_n(q)$. It means that vertices v and u are within same connected component of I_s(v) $= I_s$(u) for all possible values of s. We shall assume that I_s, $s = 1, 2, \dots, n$ are polynomial expressions in n variables over the field F_q.

Remark. The number of chosen invarians can be zero, like in case of connected graph.

We shall consider the case of a bipartite graphs $L_n(q)$, but one can modify easily the procedure below on the case of general linguistic graphs.

As in previous section the plaintext and the ciphertext are n-tuples over F_q, $q > 2$ and we identify them with points (or lines) of the graph $L_n(q)$. Alice shall choose to keep her graph secret. We may assume that the number of nonisomorphic bipartite linguistic graphs on q^n points and q^n lines is growing with n. Thus the cryptoanalist (Catherina) does not know the operator $N_c(\mathrm{v})$.

In transforming plaintext into ciphertext Alice shall work with two intermediate vectors denoted $\mathrm{u} = (u_1, \ldots, u_n) \in {F_q}^n$ and $\mathrm{v} = (v_1, \ldots, v_n) \in {F_q}^n$.

First, Alice choose the *symbolic key* $\mathrm{k} = (P_1, \ldots, P_t)$, where $P_i = P_i(x_1, \ldots x_k)$ are polynomials from $F_q[x_1, \ldots, x_k]$ of degree ≥ 0 such that $P_i \neq P_{i+2}$, $i = 0, \ldots, t-2$.

REMARK: If we are working *without graph invariants* $I_1, \ldots, I_k$, then $P_1, \ldots, P_t$ are constants from F_q. We shall use term *constant password* in this case.

In addition, Alice chooses two secret affine transformations, i.e. two invertible matrices $A = (a_{ij}), 1 \leq i, j \leq n$ and $B = (b_{i,j}), 1 \leq i, j \leq n$ with entries in F_q and the constant vectors $\mathrm{c} = (c_1, \ldots, c_n)$ and $\mathrm{d} = (d_1, \ldots, d_n)$.

The purpose of the two affine transformations is "to hide the graph and to hide the walk on the hidden graph". First, she sets $\mathrm{u} = A \times \mathrm{x} + \mathrm{c}$

Second, Alice takes the plaintext $(x_1, \ldots, x_n)$ and computes the *numerical password* $\mathrm{K} = (K_1, \ldots, K_t)$, where

$$K_s = P_s(I_1(u_1, \ldots u_n), \ldots, I_k(u_1, \ldots, u_n)).$$

Next, she would like to have $\mathrm{v} \in {F_q}^n$ simply equal to the

$\mathrm{v} = N(\mathrm{u})$, where $N(\mathrm{u}) = N_{K_t + c(\mathrm{u})}(N_{K_{t-1}}(\ldots N_{K_2}(N_{K_1}(\mathrm{u}))) \ldots)$.

Finally, Alice sets $\mathrm{y} = B^{-1}(\mathrm{v} - \mathrm{d})$ (that is $\mathrm{v} = B\mathrm{y} + \mathrm{d}$).

However, her public encryption rule will go right from x to y and shall not include the transformations N_{K_i} at all. The main part of computation of public rule is computation of the polynomial map $T : \mathrm{u} \rightarrow \mathrm{v}$ can be done with "Maple", "Matematica" or other software package for symbolic computations. After, Alice will combine T with two affine transformations and get a formula

$\mathrm{y} = (F_1(x_1, \ldots, x_n), \ldots, F_n(x_1, \ldots, x_n)$,

where $F_i(x_1, \ldots, x_n)$ are polynomials in n variables written in expanded form, i.e. as the sums of monomials of kind ${x_1}^{i_1} \ldots {x_n}^{i_n}$ with the coeficients from F_q. Alice makes polynomial equations $y_i = F_i(x_1, \ldots, x_n)$ public.

Again, like in Imai-Matsumoto scheme, if Bob wants to send her a plaintext message x, he just substitutes x_i in the public equations and finds y_i. On the other hand Catherine, who knows only the ciphertext and the public key must solve a nonlinear system for the unknowns x_i.

When Alice receives the ciphertext y, she uses her knowledge of A, B, c, d, graph $L_n(q)$ and the prepassword.

Namely, she shall compute $\mathrm{v} = B\mathrm{y} + \mathrm{d}$. Next, Alice shall compute the components of numerical password: $K_i = P_i(v_1, \dots, v_n)$ because u and v are in the same component of the graph.

The inverse to the map $N : \mathrm{u} \to \mathrm{v}$ is

$$N^{-1}(\mathrm{v}) = N_{c(\mathrm{v})-K_t}(N_{K_1}(N_{K_2}(\dots(N_{t-1}(\mathrm{U})\dots)$$

because of $K_t + c(\mathrm{u}) = c(\mathrm{v})$. Thus Alice shall use iterative process to compute $\mathrm{u} = N^{-1}(\mathrm{v})$.

Finally, she computes the plaintext $\mathrm{x} = A^{-1} \times (\mathrm{u} - \mathrm{c})$.

GENERALIZATION. Let F_q be an extension of $F_{q'}$. Then F_q be a vector space over $F_{q'}$. We shall choose a basis of this vector space and consider each vertex of the linguistic graph as a tuple over $F_{q'}$. So points (lines) form the $n \times m$ -dimensional vector space V, where $m = |F_q' : F_q|$ is the number of basic elements. It is allow us to take affine transformations $A\mathrm{x} + \mathrm{c}$ and $B\mathrm{x} + \mathrm{d}$ of V, where entries of matrices and vectors are elements in $F_{q'}$ instead of those with entries in F_q. The most important case is $q' = p$, $q = p^n, p$ is prime like in Irai-Matsumoto scheme.

PRIVATE KEY ALGORITHMS.

In case of sparse linguistic graphs, i. e. graphs such that polynomials determine formula for the neighbour of given vertex, graphs invariant, symbolic key contains "small" number of monomials, and matrices A and B are sparse (small number of entries $\neq 0$) we can use the encryption above as a private key algorithm. In that case Alice and Bob both have the information about graph, chosen graph invariants, symbolic key and two affine transformation. In the best case of sparse graph encryption (and decryption) can be done for the $lO(n)$ time, where l is the length of the symbolic key and n is the length of the ciphertext.

The advantage of such method comparing with linear encryption or some other private key algorithms is that such encryption can be resistant to attacks when adversary (Catherine) has several pairs (plaintext, ciphertext) and trying to get the key. We shall assume that matrices A and B together with sparse vectors c and d and symbolic key form the password. Catherine knows the graph, chosen graph invariants, the general scheme of encrypion. So she is trying to restore the symbolic key and control the communication between Alice and Bob.

Practically, private key algorithms serves for the encryption of sufficiently large texts. Thus we have to use rather small passwords. -We may assume that the size of password, i. e. number of monomials in symbolic key together with the number of nonzero entries in A, B, c and d is constant or linear function $an + b$ where $a << 1$.

We can say that the task of Catherine in such situation is harder than in the situation of public key encryption with the same algorithm for Alice, because in "public case" Catherine can create any pair (plaintext, ciphertext). Catherine can study the graph properties and get description of connected components, list of all graph invariants, but without knowledge about the symbolic key even reduction the problem to public key case is hard.

6. *Linguistic graphs of triangular type over F_q of high girth*

We consider some heuristic arguments on the complexity of constant encryption scheme from previous section in case the linguistic graph is public, its girth is "high" and chosen affine transformation are mutually inverting. If we "hide" the graph, then decryption shall be essentially harder.

The constructions of families of graphs of high girth are connected with studies of some well-known problems in Extremal Graph Theory (see [2]). Let $ex(v, n)$ be, as usual, the greatest number of edges (size) in a graph on v vertices, which contains no cycles C_3, C_4, ..., C_n.

From Erdös' Even Cycle Theorem and its modifications [2] it follows that

$$ex(v, 2k) \leq Cv^{1+1/k} \qquad (2)$$

where C is a positive constant.

This bound is known to be sharp precisely when $k = 2, 3$, and 5. Similar upper bound for the class of biregular bipartite graphs, which includes linguistic graphs, the reader can find in [22].

If for graphs G_i, $i = 1, \infty$ of degree l_i and girth g_i we have

$$g_i \geq \gamma \log_{l_i - 1}(v_i) \qquad (3)$$

then they form an infinite family of graphs of large girth in the sense of N. Biggs [1] (see, also [5],[7],[8],[9],[11],[12],[13] for examples of such families).

We have $\gamma \leq 2$, because of (2), but no family has been found for which $\gamma = 2$. A. Lubotzky conjectured that $\gamma \leq 4/3$.

In case of encryption algorithms for the linguistic graphs of girth k with the password of length $l, l \leq k/2$ there is not more than one pass between two vertices at a distance l and we have the following property of encryption:

(P1) two different passwords of length l produce different ciphertexts

Let G be a linguistic graph over the extension F_q of the finite field $F_{q'}$ and we are combining the "graphical" encryption procedure with two affine transformations over $F_{q'}$ (see, the end of previous section). Then the property (P1) is still in place, because our affine transformations are bijections.

Let us consider the case when chosen affine transformations are mutually invert each other. Then we have the following nice property

(P2) the plaintext and ciphertext are always different.

Proof. In case of just graphical encryption π this property holds because of absence cycles of the length $2l$, i. e. permutation π does not have invariant points. The same property is valid for transformations which are conjugate with π. □

Remark Let L be the bipartite linguistic graph of high girth and π be a graphical encryption corresponding to the pass of the odd length. Then property $P2$ is immaterial because points and lines are formally different vectors but as tuples over F_q they may coinside. Anyway there are examples of such encryption scheme with the property (P_2), in particular, graphs $D(k, q)$, k is odd, below.

Let us bound the complexity of the decryption in case when our graph $L_n(q)$ is public and the length of the password is l:

(i) the plaintext x and the ciphertext y are at the distance l,

(ii) we can identify the totality of all vertices at a distance $l \leq k/2$ with the subgraph in q-regular tree,

(iii) the decryption is equivalent to finding the pass between y and x, two vertices of q-regular tree at the distance l can be estimated by

$$q(q-1)^{l-1}CN(n)(1) \tag{4}$$

, where $CN(n)$ is the complexity of the operation of taking neighbour in the graph $L_n(q)$.

For the encryption of "potentially infinite plaintext" we shall use an infinite family of graphs $F = \{L_{n_i}(q)\}$, $i = 1, \ldots, \infty$ of increasing girth, order and $CN(n_i)$ is a complexity of taking neighbour for L_{n_i}. Let us make simple suggestion that $CN(n_i) < CN(n_{i+1})$.

Then the complexity of decryption is increasing with increasing of the length of password or increasing of the size of the plaintext. If we choose l as a linear function $l = an + b$, from the length of the plaintext n then our bound above getting exponential. Such a choice of l is possible if there is a linear from n lower bound for the girth of graphs from F, i. e. F is a *family of graphs of large girth*

Let us suppose now that the graph is unknown and somebody trying to decrypt text by solving public equations $y_i = P(x_1, \ldots, x_n)$ for x_i. This task is a classical hard problem of algebra. The system above can be investigated for $d^{O(n^2)}$ steps, where $d = d(l)$ is the maximal degree of polynomials. We can do better $(d(l)^{Cn})$ if we know that the system is consistent. If l is a depending from n then this bound is much worse that(4).

If you have a family of polynomial linguistic graph of bounded degree, you may choose the dimension n such that your correspondent could encrypt but could not decrypt and use graph encryption in "public key fashion", because we should use the gap between computations of polynomial in given point and investigation of given system of equations.

If we shall conjugate "L_{n_i}"- encryption by hidden affine transformation T then complexity of decryption is not decreasing, the problem is getting harder.

Let us assume that F is a family of triangular graphs over F_q, such that $L_{n_{i+1}}$ is defined by just adding last $n_{i+1} - n_i$ equations to the equations for L_{n_i}. If girth of graphs from F is unbounded, then natural projective limit of graphs from F is the q-regular tree. In that class of graphs we can find an infinite families of graphs of large girth.

Explicit example here is the family of graphs $D(k, q)$ [7], [8], [9].

Let q be a prime power, and let P and L be two countable infinite dimensional vector spaces over $GF(q)$. 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 [7]:

$$(p) = (p_1, p_{11}, p_{12}, p_{21}, p_{22}, p'_{22}, p_{23}, \ldots, p_{ii}, p'_{ii}, p_{i,i+1}, p_{i+1,i}, \ldots),$$

$$[l] = [l_1, l_{11}, l_{12}, l_{21}, l_{22}, l'_{22}, l_{23}, \ldots, l_{ii}, l'_{ii}, l_{i,i+1}, l_{i+1,i}, \ldots).$$

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

$$l_{11} - p_{11} = l_1 p_1$$

$$l_{12} - p_{12} = l_{11} p_1$$

$$l_{21} - p_{21} = l_1 p_{11} \qquad ((5))$$

$$l_{ii} - p_{ii} = l_1 p_{i-1,i}$$

$$l'_{ii} - p'_{ii} = l_{i,i-1} p_1$$

$$l_{i,i+1} - p_{i,i+1} = l_{ii} p_1$$

$$l_{i+1,i} - p_{i+1,i} = l_1 p'_{ii}$$

(The last four relations are defined for $i \geq 2$.) This incidence structure (P, L, I) we denote as $D(q)$. We speak now of the *incidence graph* of (P, L, I), which has the vertex set $P \cup L$ and 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. The incidence I_k is then defined by imposing the first $k-1$ incidence relations and ignoring all others. For fixed q, the incidence graph corresponding to the structure (P_k, L_k, I_k) is denoted by $D(k, q)$.

Looking at equations 7.1 we can see that $D(k, q)$ are linguistic graphs of triangular type over F_q.

Proposition 6.1. [8] Let q be a prime power, and $k \geq 2$. Then for odd k, $g(D(k, q)) \geq k + 5$

Graphs $D(k, q)$, $k \geq 2$ form first infinite family of linguistic graphs of unbounded girth.

Let us consider graph invariants for $D(k, q)$.

To facilitate notation in future results, it will be convenient for us to define $p_{-1,0} = l_{0,-1} = p_{1,0} = l_{0,1} = 0$, $p_{0,0} = l_{0,0} = -1$, $p'_{0,0} = l'_{0,0} = 1$, $p_{0,1} = p_2$, $l_{1,0} = l_1$, $l'_{1,1} = l_{1,1}$, $p'_{1,1} = p_{1,1}$, and to rewrite (5) in the form :

$$l_{ii} - p_{ii} = l_1 p_{i-1,i}$$

$$l'_{ii} - p'_{ii} = l_{i,i-1} p_1$$

$$l_{i,i+1} - p_{i,i+1} = l_{ii} p_1$$

$$l_{i+1,i} - p_{i+1,i} = l_1 p'_{ii}$$

for $i = 0, 1, 2, \ldots$

Notice that for $i = 0$, the four conditions (5) are satisfied by every point and line, and, for $i = 1$, the first two equations coincide and give $l_{1,1} - p_{1,1} = l_1 p_1$.

Let $k \geq 6$, $t = [(k + 2)/4]$, and let

$$u = (u_i, u_{11}, \cdots, u_{tt}, u'_{tt}, u_{t,t+1}, u_{t+1,t}, \cdots)$$

be a vertex of $D(k,q)$. (It does not matter whether u is a point or a line). For every r, $2 \le r \le t$, let

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

and $a = a(u) = (a_2, a_3, \cdots, a_t)$. (Here we define

$p_{0,-1} = l_{0,-1} = p_{1,0} = l_{0,1} = 0$, $p_{00} = l_{00} = -1$, $p_{0,1} = p_1$, $l_{1,0} = l_1$, $l'_{11} = l_{11}$, $p'_{1,1} = p_{1,1}$).

In [9], [10], the following statement was proved.

Proposition 6.2. . Let u and v be vertices from the same component of $D(k,q)$. Then $a(u) = a(v)$. Moreover, for any $t-1$ field elements $x_i \in GF(q)$, $2 \le t \ge [(k+2)/4]$, there exists a vertex v of $D(k,q)$ for which

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

So, Alice may choose symbolic password formed by polynomials $P_i(z_1, \ldots, z_{t-1}), i = 1, \ldots, l$ and work with the numerical password formed by

$$P_i(a_2(x_1, \ldots, x_k), \ldots, a_t(x_1, \ldots, x_k)),$$

where $(x_1, \ldots, x_k)$ is the plaintext, which is the point (line) of $D(k,q)$. Choice of the type of plaintext (point or line), leeds to different encryption procedures.

Looking at the equations of $D(k,q)$ we can see that the complexity of operation to take the neighbour of chosen colour is $O(n)$. Thus the complexity of operation to encrypt (or decrypt) for Alice shall be $O(nl)$.

We realise such encryption by computer program written in Java, the following table demonstrates its speed as function in variables l and n.

Other option is to work with the connected component of $D(k,q)$ and use constant password.

If you have a family of polynomial linguistic graph of bounded degree,

you may choose the dimension n such that your correspondent could encrypt but could not decrypt and use graph encryption in "public key fashion", because we should use the gap between computations of polynomial in given point and investigation of given system of equations.

6.1. "Toy example"-exercise

The purpose of this example is to demonstrate the encryption scheme of section 3 involving graphs $D(k,q)$. We illustrate the mechanical operations of cryptosystem, but its parameters are too small to give a serious security.

1) *The linguistic graph D(14, 127)*

It is convenient to write points (p) and lines $[l]$ in the following form
(p) $= (p_1, p_{11}, p_{12}, p_{21}, p_{22}, p'_{22}, p_{23}, p_{32}, p_{33}, p'_{33}, p_{34}, p_{43}, p_{44}, p'_{44})$
[l] $= [l_1, l_{11}, l_{12}, l_{21}, l_{22}, l'_{22}, l_{23}, l_{32}, l_{33}, l'_{33}, l_{34}, l_{43}, l_{44}, l'_{44}]$
The co-ordinates p_1 and l_1 are colours of point (p) and line $[l]$, respectively.

The girth of the graph $D(14, 127)$ is 19. Thus we have advantages of graphs of high girth if the length of the password is ≤ 9.

The graph $D(k, q)$ is defined by the first 14 equations of (5). Thus the operators to take neighbour of point (p) of the colour l_1 and neighbour of line [l] of the colour p_1 can be written in the form

$N_{l_1}((p)) =$ [l] $= [l_1, p_1 + l_1p_1, p_{12} + p_1l_{11}, p_{21} + l_1p_{11}, p_{22} + l_1p_{12}, p'_{22} + p_1l_{21}, p_{23} + p_1l_{22}, p_{32} + l_1p'_{22}, p_{33} + l_1p_{23}, p'_{33} + p_1l_32, p_{34} + p_1l_{33}, p_{43} + l_1p'_{33}, p_{44} + L_1p_{34}, p'_{44} + p_1l_{43}]$

$N_{p_1}([l]) =$ (p) $= (p_1, p_{11} - l_1p_1, p_{12} - l_1p_{11}, p_{21} - l_1p_{11}, p_{22} - l_1p_{12}, p'_{22} - l_{21}p_1, p_{23} - p1l_{22}, p_{32} - l_1p'_{22}, p_{33} - l_1p_{23}, p'_{33} - p_1l_{32}, p_{34} - p_1l_{22}p_{43} - l_1p'_{33}, p_{44} - l_1p_{34}, p'_{44} - p_1l_{43})$

We have to compute co-ordinates of the (p) and [l] in natural order by iterative process.

The following graph invariants a_2(u)$, a_3$(u)$, a_4$(u), where tuple u $=< u_1, u_{11}, \ldots u'_{44} >$ can be point ot line, determine the connected components of $D(k, q)$.

a_2(u) $= ((u'_{22} - u_{21}u_{01}) + (u_{11}u'_{11} - u_{12}u_{10}) - u_{22}$

a_3(u) $= ((u'_{33} - u_{01}u_{32}) + (u_{11}u'_{22} - u_{12}u_{21}) + (u_{22} - u'_{11} - u_{23}u_{10}) - u_{33})$,

a_4(u) $= (u'_{44} - u_{01}u_{43}) + (u_{11}u'_{33} - u_{12}u_{32}) + (u_{22}u'_{22} - u_{23}u_{21}) + (u_{33}u'_{11} - u_{34}u_{10}) - u_{44})$

2) *The symbolic key*

Let us take the following symbolic key with linear components $(z_1 + z_2 + z_3 + 1, z_1 + z_2 + 2, z_1 + z_3 + 3, z_2 + z_3 + 4, z_1 + 5, z_2 + 6, z_3 + 7, z_1 - z_2 - z_3 + 8, z_2 - z_1 - z_3 + 9)$

3) *The affine transformations*

Let us choose $A = (a_{ij})$, where $a_{ij} = 1$ is a nonzero entry iff $i - j \leq 4$, $B = A^{-1}$, c $= (1, 2, 1, 0, \ldots, 0)$, d $= -B$c.

4) *Choice of type*

Let us assume that the plaintext (x) is the point.

The information 1)- 4) allow us to determine public key encryption scheme.

$< 1 >$ you compute x$'$ $= A$x + c, where x $= (x_1, \ldots, x'_{44})$ is the "symbolic" plaintext

$< 2 >$ you can compute the numerical password np(u) for the symbolic plaintext u with the "Maple" or "Matematica" by the substitution, a_i(u) instead of z_i into the formula for the symbolic key.

$< 3 >$ you have to make a substitution of x' instead of u and get $n_p = np(\mathrm{x}')$.

$< 4 >$ consecutive application of operators $N_{l_1}(p)$ and $N_{p_1}(l)$ allow you to compute $\mathrm{v} = \mathrm{x}'^{np(\mathrm{x}')}$.

$< 5 >$ you are computing the expression

$$B\mathrm{v} + \mathrm{d} = \mathrm{P}(\mathrm{x}) = (P_1(\mathrm{x}), \ldots, P'_{44}(\mathrm{x})).$$

$< 6 >$ finally you can get your public expression $\mathrm{P}^1 27(\mathrm{x})$ by taking all integer coefficients mod127

Try to invert the public polynomial map directly by the symbolic program in "Maple" or "Matematica".

You may repeat the exercise above with the constant password

$$(1, 2, 3, 4, 5, 6, 7, 8, 9).$$

References

[1] N.L. Biggs, *Graphs with large girth*, Ars Combinatoria, 25C (1988), 73–80.

[2] B. Bollobás, *Extremal Graph Theory*, Academic Press,

[3] Neal Coblitz, *A Course in Number Theory and Cryptography*, Second Edition, Springer, 1994, 237 p.

[4] Neal Coblitz, *Algebraic Aspects of Cryptography*, Springer, 1998, 198 p.

[5] W. Imrich, *Explicit construction of graphs without small cycles*, Combinatorica **2** (1984) 53–59.

[6] Imai, Matsumoto,*Public quadratic polynomial tuples for efficient signature verification and message encryption*, Advances in Cryptology, Eurocrypt '88, Springer Verlag, 419-453.

[7] F. Lazebnik and V. Ustimenko, *Some Algebraic Constructions of Dense Graphs of Large Girth and of Large Size*, DIMACS series in Discrete Mathematics and Theoretical Computer Science, V. 10 (1993), 75-93.

[8] F. Lazebnik F. and V. Ustimenko, *Explicit construction of graphs with an arbitrary large girth and of large size*, Discrete Appl. Math. , 60, (1995), 275 - 284.

[9] 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.

[10] F. Lazebnik, V. A. Ustimenko and A. J. Woldar, *A characterization of the components of the graphs $D(k, q)$*, Discrete Mathematics, 157 (1996), 271-283.

[11] A. Lubotsky, R. Philips, P. Sarnak, *Ramanujan graphs*, J. Comb. Theory., 115, N 2., (1989), 62-89.

[12] G. A. Margulis, *Explicit construction of graphs without short cycles and low density codes*, Combinatorica, 2, (1982), 71-78.

[13] G. Margulis, *Explicit constructions of concentrators*, Probl. of Inform. Transm., 10, (1975), 325-332.

[14] B. Mortimer, *Permutation groups containing affine of the same degree*, J. London Math. Soc., 15, N3, 445-455.

[15] J. Patarin, *Cryptoanalysis of the Matsumoto and Imai public key scheme of the Eurocrypt '88*, Advances in Cryptology, Eurocrypt '96, Springer Verlag, 43-56.

[16] V. A. Ustimenko, *Random walks on special graphs and Cryptography*, Amer. Math. Soc. Meeting, Loisville, March, 1988.

[17] V. A. Ustimenko, *Coordinatization of regular tree and its quotients*, In the volume "Voronoi's Impact in Modern Science": (Proceedings of Memorial Voronoi Conference, Kiev, 1998), Kiev, IM AN Ukraine, July, 1998, pp. 125 - 152.

[18] V. Ustimenko, D. Sharma, *Special Graphs in Cryptography* in Proceedings of 2000 International Workshop on Practice and Theory in Public Key Cryptography (PKC 2000), Melbourne, December 1.

[19] V. Ustimenko and D. Sharma, *CRYPTIM: The system to encrypt text and image data*, in Proceedings of International ICSC congress on Intelligent Systems and Applications, December 2000, University of Wollongong, 14 pp.

[20] V. Ustimenko, *CRYPTIM: Graphs as Tools for Symmetric Encryption*, in Lecture Notes in Computer Science, Springer, v. 2227, 278-287.

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

[22] V. Ustimenko, A. Woldar, *Extremal properties of regular and affine generalised polygons of tactical configurations*, 2003, European Journal of Combinatorics, 2003, v. 24 , 99-111.

[23] V. Ustimenko, Yu. Khmelevsky, *Walks on graphs as symmetric and assymetric tools for encryption*, South Pacific Journal of Natural Studies, 2002, vol. 20, 23-41.

[24] V. Ustimenko, A. Tousene, *CRYPTALL-the system encrypt all type of data* (to appear)

[25] A. L. Weiss, *Girth of bipartite sextet graphs*, Combinatorika 4 (2-3) 1984, 241-245.

CONTACT INFORMATION

V. A. Ustimenko Kiev Mohyla Academy (Ukraine)
E-Mail: `vau@ukma.kiev.ua`

Received by the editors: 22.02.2004
and in final form 21.03.2005.