ALBANIAN JOURNAL
OF MATHEMATICS
Volume 2, Number 3, Pages 229–234
ISSN 1930-1235: (2008)

ON SOME PROPERTIES OF GRAPH BASED PUBLIC KEYS

ANETA WROBLEWSKA

Maria Curie-Skłodowska University
Lublin, Poland.
e-mail: wroblewska-aneta@wp.pl

Abstract. In this paper we will evaluate degrees of nonlinear polynomial encryption transformation in F_q^n, which was defined in [11] in terms of the walk on the graph with vertex set $F_q^n \cup F_g^n$. Independently from the length of the walk, this transformation has degree 3. It means that public user can do the encryption process in polynomial time.

1. Introduction

We will study some properties of graph base public key algorithm, which was proposed in [11]. Some generalization of this method the reader can find in [13], [14]. First (Section 2) we introduce some definitions needed to describe our algorithm. Like in the well known example of polynomial encryption used by Imai and Matsumoto or Patarin in his "small dragon" ([6], [7]) in Section 3 we combine "graphical encryption" P with two affine transformations T_1 and T_2 and work with the public map $Q = T_1PT_2$. After such transformation we get a system of polynomial map and in Section 4 we will investigate its degrees in order to find out a heuristic complexity of this cryptosystem.

Let us use traditional characters in Cryptography: Alice is the holder of the key, Bob — the public user and the cryptoanalyst — Catherine ([5], [6]). The speed of the software implementation of symmetric encryption algorithm for Alice is evaluated in [14]. Evaluation of the degree of the transformation demonstrated the feasibility of the algorithm for Bob.

2. Graph based encryption algorithm

We define graph as an irreflexive and symmetric binary relation $\phi \subset V \times V$, where V is the set of vertices. Missing graph theoretical definition can be find in [1], [2]. In this subsection we will consider the *parallelotopic graphs* and linguistic graph with alphabet $M = GF(q)$. Here, our messages (plaintexts or ciphertexts) and enryption tools (passwords) are tuples over $GF(q)$. What is important the idea can be expand to arbitrary chosen commutative ring K, which leads to a very fast cryptoalgorithm (operation in $K = Z_{p^n}$ are much faster than in case of F_{p^n} for large n).

We say 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), $v \in V(\Gamma)$,

Key words and phrases. Public Key Cryptography, polynomial encryption.

$m \in M$, there is a unique neighbour u of v satisfying $\pi(u) = m$. For given vertex v, and any colour m, there exists exactly one neighbour u of v of colour m. Then the neighborhood of each vertex looks like rainbow ie. consist of $|M|$ vertices of different colours. This is obvious that the graph is k-regular with $k = |M|$.

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 of a vertex v with colour t. Then the password be the string of colours $(t_1, t_2, \ldots, t_n)$, $t_i \in M$ such that $t_i \neq t_{i+2}$ and encryption process is the composition $N_{t_1} \times N_{t_2} \times \cdots N_{t_n}$ of bijective maps $N_{t_i} : V(\Gamma) \rightarrow V(\Gamma)$. If the plaintext $v \in V(\Gamma)$ is given, then the encryption procedure corresponds to the followin chain in the graph: $v \rightarrow v_1 = N(t_1, v) \rightarrow v_2 = N(t_2, v_1) \rightarrow \cdots \rightarrow v_n = N(t_n, v_{n-1})$. It is clear that $(t_{n-1}, \cdots, t_1, c(v))$ is the "decoding tuple", because it corresponds to the decoding arc.

We use the term linguistic graph over $GF(q)$ when we have a linguistic graph with alphabet $M = GF(q)$ and the set of neighbors 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.

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:

$$(p) = (p_1, p_{1,1}, p_{1,2}, p_{2,1}, p_{2,2}, p'_{2,2}, p_{2,3}, p_{3,2}, p_{3,3}, p'_{3,3}, \ldots)$$

$$[l] = [l_1, l_{1,1}, l_{1,2}, l_{2,1}, l_{2,2}, l^{'}_{2,2}, l_{2,3}, l_{3,2}, l_{3,3}, l^{'}_{3,3}, \ldots]$$

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 coordinates hold:

$$l_{1,1} - p_{1,1} = l_1 p_1$$

$$l_{1,2} - p_{1,2} = l_{1,1} p_1$$

$$l_{2,1} - p_{2,1} = l_1 p_{1,1}$$

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

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

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

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

To decrypt the data we use a key or password of length m, $\rho = (\alpha_1, \ldots, \alpha_m)$, where α_i's are password characters such as α_i is different from α_{i+2}. Each arc of the graph represents one possible character. For the plaintext $(p_1, \ldots, p_n)$, in each number of walk j, l_1 or p_1 will be $p_1 + \alpha_1 \cdots + \alpha_j$.

3. Linguistic graphs system hidden by the affine transformation.

Let F_q, $q > 2$ be the finite field, where q is a prime power. Alice shall be using the hidden graph scheme based 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. 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.

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 constant password $\alpha = \alpha_1\alpha_2 \ldots \alpha_n$. In addition, Alice chooses two secret affine transformations, i.e. two invertible matrices $A = (a_{ij}), 1 \le i, j \le n$ and $B = (b_{i,j}), 1 \le i, j \le n$ with entries in F_q and the constant vectors $\mathrm{c} = (c_1, \ldots, c_n)$ and $\mathrm{d} = (d_1, \ldots, d_n)$. We use the two affine transformations in order to hide the graph and to hide the walk on the hidden graph. Then, she sets $\mathrm{u} = A \times \mathrm{x} + \mathrm{c}$. Next, she would like to have $\mathrm{v} \in {F_q}^n$ simply equal to the $\mathrm{v} = N(\mathrm{u})$, received from graph based algorithm. Finally, Alice sets $\mathrm{y} = B^{-1}(\mathrm{v} - \mathrm{d})$ (that is $\mathrm{v} = B\mathrm{y} + \mathrm{d}$). 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 $A, B, \mathrm{c}, \mathrm{d}$, graph $L_n(q)$ and the password. Namely, she shall compute $\mathrm{v} = B\mathrm{y} + \mathrm{d}$. Then Alice using iterative process of decryption based on the graph can compute $\mathrm{u} = N^{-1}(\mathrm{v})$. Finally, she computes the plaintext $\mathrm{x} = A^{-1} \times (\mathrm{u} - \mathrm{c})$.

4. Degrees of polynomials in ciphertext

Before applying affine transformation we want to find out a degree of polynomial map $T : u \to v$. We take the password $\alpha = \alpha_1\alpha_2 \ldots \alpha_n$ which is used by adding element α_j to the first character of the encrypted data in each walk number j. Therefore we are getting transformation $T_{\alpha_1}T_{\alpha_2} \ldots T_{\alpha_n}$. If we treat the elements of the plain data before encryption as variables, in each transformation T_{α_1}, $T_{\alpha_1}T_{\alpha_2}$, $T_{\alpha_1}T_{\alpha_2} \ldots T_{\alpha_n}$ we get a polynomials of these variables. We would like to find out a degree of the polynomials.

4.1. **Transformation T_{α_1}.** Our research we start with studying transformation T_{α_1}. Hence we have:

$$l_1 = p_1 + \alpha_1 \qquad \deg l_1 = 1$$

$$l_{1,1} = p_{1,1} + l_1 p_1 = p_{1,1} + \alpha_1 p_1 + p_1^2$$

$$l_{1,2} = p_{1,2} + p_1 l_{1,1} = p_{1,2} + p_1 p_{1,1} + \alpha_1 p_1^2 + p_1^3$$

$$l_{i,i} = p_{i,i} + l_1 p_{i-1,i} = p_{i,i} + \alpha_1 p_{i-1,i} + p_1 p_{i-1,i}$$

$$l_{i,i+1} = p_{i,i+1} + p_1 l_{i,i} = p_{i,i+1} + \alpha_1 p_1 p_{i-1,i} + p_1 p_{i,i} + p_1^2 p_{i-1,i}.$$

Similarly we are receiving:

$$l_{i+1,i} = p_{i+1,i} + l_1 p^{'}_{i,i} = p_{i+1,i} + \alpha_1 p^{'}_{i,i} + p_1 p^{'}_{i,i}$$

$$l'_{i,i} = p'_{i,i} + p_1 l_{i,i-1} = p'_{i,i} + \alpha_1 p_1 p'_{i-1,i-1} + p_1 p_{i,i-1} + p_1^2 p_{i-1,i-1}.$$

So if we take the plane data (p) as $(p_1, p_2, \dots, p_n)$ after this transformation we get the line vertex $f_1(p_1), f_2(p_1, p_2), \dots, f_n(p_1, p_2, \dots, p_n)$,

$$\deg f_n(p_1, p_2, \dots, p_n) = \begin{cases} 1, & n = 1, \\ 2, & n = 2, \\ 2, & n = 4k, 4k+1, \\ 3, & n = 4k+2, 4k-1 \quad \text{where} \quad k = 1, 2, 3 \dots \end{cases}$$

4.2. **Transformation** $T_{\alpha_1} T_{\alpha_2}$**.** Using the previous part of the calculation (transformation T_{α_1}) we can calculate elements of the encrypted data after transformation T_{α_2}.

$$p_1^{(2)} = p_1 + \alpha_1 + \alpha_2$$

$$p_{1,1} = l_{1,1} - l_1 p_1^{(2)} = -(\alpha_1 + \alpha_2)(\alpha_1 + p_1)$$

$$p_{1,2}^{(2)} = l_{1,2} - p_1^{(2)} l_{1,1} = p_{1,2} - (\alpha_1 + \alpha_2) p_{1,1} - \alpha_1(\alpha_1 + \alpha_2) p_1 - (\alpha_1 + \alpha_2) p_1^2$$

$$p_{i,i+1}^{(2)} = l_{i,i+1} - p_1^{(2)} l_{i,i} = p_{i,i+1} - (\alpha_1 + \alpha_2)(p_{i,i} + \alpha_1 p_{i-1,i} + p_1 p_{i-1,i})$$

$$p_{i,i}^{(2)} = l_{i,i} - l_1 p_{i-1,i}^{(2)} = p_{i,i} + (\alpha_1 + p_1)(\alpha_1 + \alpha_2)(p_{i-1,i-1} + \alpha_1 p_{i-2,i-1} + p_1 p_{i-2,i-1})$$

Similarly we are receiving:

$$p_{i,i}^{\prime(2)} = l'_{i,i} - p_1^{(2)} l_{i,i-1} = p^{'}_{i,i} - (\alpha_1 + \alpha_2)(p_{i,i-1} + \alpha_1 p_{i-1,i-1} + p_1 p'_{i-1,i-1})$$

$$p_{i+1,i}^{(2)} = l_{i+1,i} - l_1 p_{i,i}^{\prime(2)} = p_{i+1,i} + (\alpha_1 + p_1)(\alpha_1 + \alpha_2)(p_{i-1,i-1} + \alpha_1 p'_{i-1,i-1} + p_1 p'_{i-1,i-1})$$

Hence we got vertex point $(p) = (g_1(p_1), g_2(p_1, p_2), \dots, g_n(p_1, p_2, \dots, p_n))$ and degrees of each component are following:

$$\deg g_n(p_1, p_2, \dots, p_n) = \begin{cases} 1, & n = 1, \\ 1, & n = 2, \\ 2, & n = 4k-1, 4k+2, \\ 3, & n = 4k, 4k+1 \quad \text{where} \quad k = 1, 2, 3 \dots \end{cases}$$

4.3. **Transformation** $T_{\alpha_1} T_{\alpha_2} \dots T_{\alpha_m}$**.** Degrees of elements of vertex point and vertex line after transformation $T_{\alpha_1} T_{\alpha_2} \dots T_{\alpha_{m-1}}$ and $T_{\alpha_1} T_{\alpha_2} \dots T_{\alpha_m}$ respectively, we will calculate using induction, imposing m-even.

Assume transformation $T_{\alpha_1} T_{\alpha_2} \dots T_{\alpha_{m-3}}$ gave us vertex point:

$$(p)^{(m-3)} = (g_1^{(m-3)}(p_1), g_2^{(m-3)}(p_1, p_2), \dots, g_n^{(m-3)}(p_1, p_2, \dots, p_n))$$

with degree

$$\deg g_n^{(m-3)}(p_1, p_2, \dots, p_n) = \begin{cases} 1, & n = 1, \\ 1, & n = 2, \\ 2, & n = 4k-1, 4k+2, \\ 3, & n = 4k, 4k+1 \quad \text{where} \quad k = 1, 2, 3, \dots \end{cases}$$

and vertex line after transformation $T_{\alpha_1} T_{\alpha_2} \dots T_{\alpha_{m-2}}$:

$$[l]^{(m-2)} = (f_1^{(m-2)}(p_1), f_2^{(m-2)}(p_1, p_2), \dots, f_n^{(m-2)}(p_1, p_2, \dots, p_n))$$

with degree

$$\deg f_n^{(m-2)}(p_1, p_2, \ldots, p_n) = \begin{cases} 1, & n = 1, \\ 2, & n = 2, \\ 2, & n = 4k, 4k+1, \\ 3, & n = 4k+2, 4k-1 \quad \text{where} \quad k = 1, 2, 3, \ldots \end{cases}$$

Now we have to check the degree of polynomial $g_n^{(m-1)}$.

$$\begin{aligned} p_1^{(m-1)} &= p_1 + \alpha_1 + \alpha_2 + \ldots + \alpha_{m-3} + \alpha_{m-2} + \alpha_{m-1} \\ &= p_1^{(m-3)} + \alpha_{m-2} + \alpha_{m-1} \\ p_{i,i+1}^{(m-1)} &= l_{i,i+1}^{(m-2)} - p_1^{(m-1)} l_{i,i}^{(m-2)} \\ &= p_{i,i+1}^{(m-3)} + p_1^{(m-3)} l_{i,i}^{(m-2)} - p_1^{(m-3)} l_{i,i}^{(m-2)} - (\alpha_{m-2} + \alpha_{m-1}) l_{i,i}^{(m-2)} \\ &= p_{i,i+1}^{(m-3)} - (\alpha_{m-2} + \alpha_{m-1}) l_{i,i}^{(m-2)} \end{aligned}$$

Since $p_{i,i+1}^{(m-3)}$ is independent from α_{m-2} and α_{m-1} and both $p_{i,i+1}^{(m-3)}$ and $l_{i,i}^{(m-2)}$ have degree equal 2, we get that $p_{i,i+1}^{(m-1)}$ has degree 2.

By similar reasoning we obtain that $p_{i,i}^{(m-1)}$ has degree 3, $p_{i,i}^{'(m-1)}$ degree 2, $p_{i+1,i}^{(m-1)}$ degree 3.

Hence by means of transformation $T_{\alpha_1} T_{\alpha_2} \ldots T_{\alpha_{n-1}}$ we encoded plain text $(p_1, p_2, \ldots, p_n)$ on ciphertext

$$(p)^{(m-1)} = (g_1^{(m-1)}(p_1), g_2^{(m-1)}(p_1, p_2), \ldots, g_n^{(m-1)}(p_1, p_2, \ldots, p_n))$$

with degree

$$\deg g_n^{(m-1)}(p_1, p_2, \ldots, p_n) = \begin{cases} 1, & n = 1, \\ 1, & n = 2, \\ 2, & n = 4k-1, 4k+2, \\ 3, & n = 4k, 4k+1 \quad \text{where} \quad k = 1, 2, 3, \ldots \end{cases}$$

In the same way using second part of inductive assumption we get the ciphertext $[l]^{(m)} = (f_1^{(m)}(p_1), f_2^{(m)}(p_1, p_2), \ldots, f_n^{(m)}(p_1, p_2, \ldots, p_n))$ after transformation $T_{\alpha_1} T_{\alpha_2} \ldots T_{\alpha_m}$ with

$$\deg f_n^{(m)}(p_1, p_2, \ldots, p_n) = \begin{cases} 1, & n = 1, \\ 2, & n = 2, \\ 2, & n = 4k, 4k+1, \\ 3, & n = 4k+2, 4k-1 \quad \text{where} \quad k = 1, 2, 3, \ldots \end{cases}$$

5. Remarks on the complexity of public rules

Using previous subsections, combining graph transformation T with two affine transformation, Bob get a formula:

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

where $F_i(x_1, \ldots, x_n)$ are polynomials of n variables written as the sums of monomials of kind $x_{i_1} \ldots x_{i_3}$, where $i_1, i_2, i_3 \in 1, 2, \ldots, n$ with the coefficients from F_q. Hence the polynomial equations $y_i = F_i(x_1, x_2, \ldots, x_n)$, which are made public, have degree 3. Hence the process of straightforward encryption can be done in polynomial time $O(n^4)$ (to compute one $y_i,\ i = 1, 2 \ldots, n$ we need not more than $3n^3 + n^3$ additions and multiplications). But the cryptanalyst Catherine, having a only a formula for y, has very hard task to solve the system of n equations in n variables of degree 3. We know that the variety of solution has the dimension 0. So general algorithm for such mass problem has exponential time $3^{O(n)}$ (different versions the reader can find in [3], [4], [9], [10]).

But of course for our specific system faster algorithm may exist. We encourage cryptanalysts to make an effort to break the cryptosystem.

References

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

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

[3] B. Buchberger, *Grobner Bases: An Algorithmic Method in Polynomial Ideal Theory* , Recent Trends in Multidimentional Systems Theory, N.K.Bose ed., D.Reidel Publishing comp., 1983, 184232.

[4] J. Canny *Generalized characteristic polynomials*, J. Symbolic Computation, 1990, No. 9, 241-250.

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

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

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

[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] B. Mourrain, *Bezoutian ande quotient ring structure* J. of Symbolic Computations, 39 (2005), 397-415.

[10] T. R. Seifullin, *Determination of the basis of the space of all root functionals of a system of polynomial equations and the basis of its ideal by the operation of extension of bounded root functionals* (Russian) Dopov. Nats. Akad. Nauk Ukr., Mat. Prirodozn. Tekh. Nauki 2003, No.8, 29-36 (2003)

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

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

[13] V. Ustimenko, *On the extremal graph theory for directed graphs and its cryptographical applications* In: T. Shaska, W.C. Huffman, D. Joener and V. Ustimenko, Advances in Coding Theory and Cryptography, Series on Coding and Cryptology, vol. 3, 181-200 (2007).

[14] V. Ustimenko, *On the graph based cryptography and symbolic computations*, Serdica Journal of Computing, Proceedings of International Conference on Application of Computer Algebra, ACA-2006, Varna, N1 (2007).