Proceedings of the International Multiconference on Computer Science and Information Technology pp. 297–301
ISBN 978-83-60810-14-9
ISSN 1896-7094

On the hidden discrete logarithm for some polynomial stream ciphers

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

***Abstract*—The paper is devoted to the special key management algorithm for the stream ciphers defined in [12] via finite automata corresponding to the family of directed algebraic graphs of high girth and two affine transformation. Security of the key based on the complexity of the discrete logarithm problem. It has additional heuristic security because of the "hidden base" and "hidden value" of the discrete logarithm function. We consider the heuristic evaluation of the complexity for different attack by the adversary on our private key cipher. The detailed description of the whole algorithm is given. Implemented software package has been used for the evaluation of mixing properties and speed of the private key encryption.**

I. Introduction

WE WILL introduce the key management block for the stream ciphers defined in [12] via finite automata corresponding to the family of directed algebraic graphs of high girth. The time evaluation of software package implementing these algorithms compares well with the performance of fast but not very secure RC4, DES, algorithms based on simple graphs (symmetric anti reflexive binary relations) developed during last ten years [4].

In Section 3 we described modified algorithm in term of arithmetical dynamical systems. We add the key management block to our algorithm and consider the heuristic evaluation of active attacks by the adversary.

The explicit description of the dynamical system is given in last section in terms of corresponding finite automata. We give the explicit formula for invariants we use in algorithm.

Section 4 devoted to evaluation of speed of the encryption in case of special rings of kind Z_2^s, $s \geq 8$.

II. Basic Cryptographical Terminology

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

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

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

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

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

(ii) *known plaintext*—the adversary has some plaintextx and corresponding ciphertexts.

Nowadays the security of the plaintext rests on encryption algorithm (or private key algorithm), depended on chosen key (password), which has good resistance to attacks of type (i) and (ii), and algorithm for the key exchange (public keys) with good resistance to active attacks, when the adversary can generate each plaintext p and get the corresponding plaintext c (see [2], [3], [9] or [10]). The combination of appropriate private key and public key algorithms can lead to symmetric algorithm with good resistance even to active attacks. The example of such combination will be given in the next section of the paper.

III. Arithmetical Dynamical Systems as Encryption Tools

Let K be the commutative ring, $F(K) = K[t, x_1, x_2, \dots]$ is the ring of all polynomials in variables $t, x_1, x_2, \dots$. We use symbol $\mathrm{Reg}(K)$ for the totality of regular elements i.e not a zero divisors: $a \in \mathrm{Reg}(K)$ implies $a \times x \neq 0$ for each $x \neq 0$. Let $K^\infty = \{\mathrm{x} = (t, x_1, x_2, \dots) | x_i \in K, t \in K, \mathrm{supp(x)}, \infty\}$ and $K^n = \{(x_1, x_2, \dots, x_n) | x_i \in K\}$.

Let us consider two polynomial maps P and R of K^∞ into K^∞:

$$(t, x_1, x_2, \dots,) \to (t, P_1(t, x_1, x_2, \dots), P_2(t, x_1, x_2, \dots), \dots)$$

and

$$(t, x_1, x_2, \dots,) \to (t, R_1(t, x_1, x_2, \dots), R_2(t, x_1, x_2, \dots), \dots),$$

where $P_i(t, x_1, x_2, \dots)$ and $R_i(t, x_1, x_2, \dots)$, $i = 1, 2, \dots$ are elements of $F(K)$.

We consider two families: ${f_t}^n$ and ${g_t}^n$ of K^n onto K^n sending $(x_1, x_2, \dots, x_n)$ to

$$(P'_1(t, x_1, x_2, \dots), P'_2(t, x_1, x_2, \dots), P'_n(t, x_1, x_2, \dots x_n))$$

and

$$(R'_1(t,x_1,x_2,\dots),R'_2(t,x_1,x_2,\dots),R'_n(t,x_1,x_2,\dots x_n)),$$

where P'_i and R'_i, $i=1,2,\dots,n$ correspond to the specialisations $x_{n+1}=0, x_{n+2}=0,\dots$ of P_i and R_i associated with the pair (P, R). We identify f_t and g_t, $t\in K$ with the corresponding maps $K^n\to K^n$

Let $\mathrm{r}=(r_1,r_2,\dots r_t)\in \mathrm{Reg}(K)^t$ be the tuple of length $l(\mathrm{r}))=t$. We introduce F_r, as the composition of maps $f_{r_1},g_{r_2},\dots,f_{r_{2s-1}},g_{r_{2s}}$ in case of $t=2s$ and as composition of $f_{r_1},g_{r_2},\dots,f_{r_{2s-1}},g_{r_{2s}},f_{r_{2s+1}}$ for $t=2s+1$.

We say that the pair P and R form an arithmetical dynamical system depending on time t if the following conditions hold

1) existence of $\mathrm{x}=(x_1,\dots,x_n)\in K^n$ such that $f_{t_1}(x_1,x_2,\dots,x_n)=f_{t_2}(x_1,x_2,\dots,x_n)$ for some t_1 and t_2 implies the equality $t_1=t_2$.

2) maps f_t and g_t, $t\in K$ are bijections and f_{-t} and g_{-t} are inverse maps for them.

3) There is a constant c, $c>o$ such that for each pair of tuples r, b of regular elements, conditions $l(\mathrm{r})\le cn$, $l(\mathrm{r})\le cn$ and $F_\mathrm{r}(\mathrm{x})=F_\mathrm{b}(\mathrm{x})$ for some x implies $\mathrm{r}=\mathrm{b}$.

If (P,R) form an arithmetical dynamical system, then the inverse of F_r, $l(\mathrm{r})=2s+1$ is F_b, where

$$\mathrm{b}=(-r_{2s+1},-r_{2s},\dots,-r_1).$$

If $l(\mathrm{r})=2s$ then F_r^{-1} is the composition of $g_{-r_{2s}}$ and F_d, where $\mathrm{d}=(-r_{2s-1},-r_{2s-2},\dots,-r_1)$.

We can treat K^n as the plainspace, refer to the union U of $\mathrm{Reg}(K)^t$, $1<t<cn$ as the key space and treat $\mathrm{x}\to F_\mathrm{a}(\mathrm{x})$ as the encryption map corresponding to the key a. The ciphertext $y=F_\mathrm{a}(\mathrm{x})$ can be decrypted by the map F_a^{-1} written above. So the algorithm is symmetrical. The property 3 implies that different keys of length $<cn$ produce distinct ciphertexts.

We introduce the following directed graph $\phi=\phi(n)$ corresponding to maps ${f_t}^n$ and ${g_t}^n$ over K^n. Firstly we consider two copies of P (set of points) and L (set of lines) of the free module K^n. We connect point $p\in P$ with the line $l\in L$ by directed arrow if and only if there is $t\in\mathrm{Reg}(K)$ such that $f_t(p)=l$. Let t be the colour of such a directed arrow. Additionally we join $l\in L$ and $p\in P$ by directed arrow with the colour t if there is $t\in\mathrm{Reg}(K)$ such that $g_t(l)=p$. We can consider ϕ as finite automaton for which all states are accepting states. We have to chose point p (plaintext) as initial state. It is easy to see that f_t and g_t are the transition functions of our automaton. Let $t_1,\dots,t_s$ be the "program" i.e. sequence of colours from $\mathrm{Reg}(K)$. Then the computation is the directed pass p, $f_{t_1}(p)=p^1, g_{t_2}(p^1)=p^2,\dots$. If s is even then the last vertex is $f_{t_s}(p^{s-1})$, in case of odd s we get $g_{t_s}(p^{s-1})=p^s$ as the result of the computation (encryption). The stop of the automata corresponds just to the absence of the next colour.

The inverse graph ${\phi(n)}^{-1}$ can be obtained by reversing of all arrows in ϕ. We assume that colours of arrow in ϕ and its reverse in ϕ^{-1} are the same. So we can consider ${\phi(n)}^{-1}$ as an automaton as well. Then the decryption procedure starting from the ciphertext p^s corresponds to the pass in ϕ^{-1} defined by sequence of colours $-t_s,-t_{s-1},\dots,-t_1$.

Finally, we can consider well defined projective limit ϕ of automata ϕ^n, $n\to\infty$ with the transition function $P_t(x_1,x_2,\dots)=P(t,x_1,x_2,\dots)$ and $R_t(x_1,x_2,\dots)=R(t,x_1,x_2,\dots)$. In case of finite K we can use ϕ as a Turing machine working with the potentially infinite text in the alphabet K. Results of [41] allow to formulate the following statement.

THEOREM 1

For each commutative ring K there are a cubical polynomial maps P and R on K^∞ forming arithmetical dynamical system with the constant $c\ge 1/2$ such that for each string r *of elements from* $\mathrm{Reg}(K)$ *the polynomial map F_r is cubical.*

The example as above has been defined explicitly in [5] in graph theoretical terms. The maps P and R will stand further for that particular example. Corresponding to (P,R) graphs $\phi(n)$ are strongly connected i.e. from the existence of directed pass from vertex v to w follows that w and v are connected by a directed pass. So connected components of $\phi(n)$ are well defined.

We combine the encryption process F_r corresponding to finite automaton $\phi(n)$ and string r of elements from $\mathrm{Reg}(K)$ with two invertible sparse affine transformation Af_1 and Af_2 and use the composition $\mathrm{Af}_1\times F_\mathrm{r}\times\mathrm{Af}_2$ as encryption map. We refer to such a map as *deformation* of F_r. In case of $\mathrm{Af}_1={\mathrm{Af}_2}^{-1}$ we use term *desynchronization*. In case of desynchronization the ciphertext is always distinct from the plaintext. We assume that Af_1 and Af_2 are parts of the key. Deformated or desynchronised encryption is much more secure, because it prevents adversary to use group automorphisms and special ordering of variables during his/her attacks.

In the case of deformation with fixed Af_1 and Af_2 and flexible r the property that the different passwords of kind r lead to different ciphertexts is preserved, but the situation, where the plaintext and corresponding ciphertext are the same can happen. Anyway the probability of such event is $1/|V|$, where $V=K^n$ is the plainspace.

A. Watermarking Equivalence and Hidden Discrete Logarithm

THEOREM 2

Let $\phi(n)$, $n\ge 6$ be the directed graph with the vertex set K^{k+1} defined above for the pair (P,R).

(i) There are the tuple $a=a(\mathrm{x})$, $\mathrm{x}\in K^{n+1}$ of quadratic polynomials $a_2,a_3,\dots,a_t$, $t=[(n+2)/4]$ in $K[x_0,x_1,\dots,x_n]$ such that for each directed pass $u=v_0\to v_1\to v_n=v$ we have $a(u)=a(v)$.

(ii) For any $t-1$ ring elements $x_i\in K$), $2\le t\le[(k+2)/4]$, there exists a vertex v of $\phi(n)$ for which $a(v)=(x_2,\dots,x_t)=(x)$. So classes of equivalence relation $\tau=\{(u,v)|a(u)=a(v)\}$ are in one to one correspondents with the tuples in K^t.

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

We refer to τ as watermarking equivalence and call C as above generalised connected component of the graph,

Let $|K| = d$ and η numerating function i.e bijection between K and $\{0, 1, \ldots, d-1\}$. For each tuple $\mathrm{t} = (t_0, t_1, \ldots t_s) \in K^{s+1}$ we consider its number $\eta(\mathrm{t}) = \eta(t_0) + \eta(t_1)d + \cdots + \eta(t_s)d^s$. Let $\mathrm{Reg}(K) = b \geq 2$, μ be the bijection between $\mathrm{Reg}(K)$ and $\{0, 1, \ldots, b-1\}$. We obtain $\mathrm{reg}(\mathrm{t})$ by taking the string of digits for $\eta(\mathrm{t})$ base b and computing μ^{-1} for each digit. So $\mathrm{reg}(\mathrm{t})$ is a string of characters from the alphabet $\mathrm{Reg}(K)$.

THE ALGORITHM: Correspondents Alice and Bob are taking smallest prime number p from interval $(b^{[(n+5)/2c]}, b^{[(n+5)/2]})$, where c is some constant $> 3/2$ and some number m, $m < p$. Alice takes the plainspace x computes string $a(\mathrm{x})$ (see theorem ?), then $z = \eta(a(\mathrm{x}))$ and $u = z^m \bmod p$. She treats u as integer and takes string $d(\mathrm{x}) = \mathrm{reg}(u)$ of characters from $\mathrm{Reg}(K)$. Her encryption is $\mathrm{Af}_1 \times F^n_{d(\mathrm{x}} \times \mathrm{Af}_2$. We think that numbers m, c and fixed maps Af_i, $i = 1, 2$ are parts of the key.

Let c be the ciphertext. Bob computes z defined above az $z = \eta(a(\mathrm{c}))$, computes string $d(\mathrm{x})$ and use decryption map $(\mathrm{Af}_2)^{-1} \times {F^n_{d(\mathrm{x})}}^{-1} \times \mathrm{Af}_1$.

Let $C_n(\mathrm{x}) = C(\mathrm{x})$ be the encryption function corresponding to deformation of dynamical system. The adversary may try to find the factorization $C_n(\mathrm{x}) = ((\mathrm{Af}_1)) \times F^n_{d(\mathrm{x})} \times \mathrm{Af}_2$, where $Af_i, i = 1, 2$ are unknown and the function $d(\mathrm{x})$ is $\mathrm{reg}((\eta(a(\mathrm{x})^m))$, where m is unknown also. During his active attack he can compute finite number of values $C(\mathrm{x}_i)$, $i \in J$ and use this information for finding the factorization. The following heuristic argument demonstrate that it can be difficult task.

Let as assume that affine transformation Af_i, $i = 1, 2$ are known for adversary. Notice that finding them can be very difficult. Then the adversary can compute $\mathrm{d}_i = F^n_{d(\mathrm{x}_i)} = (\mathrm{Af}_1)^{-1} C(\mathrm{x}_i)(\mathrm{Af}_2)^{-1}$. The pass between vertices of the graph is unique. The Dijkstra algoritm is not suitable for finding the pass because the vertex space of the graph is the plainspace. But may be large group of automorphisms (see [14] and further referenes) will allow to find the pass. Then the adversary computes number $b_i = \eta(a(\mathrm{x})^m$ modulo known big prime. Still he is not able to find number m because of the complexity of discrete logarithm problem. So he has to take for the set $\{\mathrm{x}_i | i \in J\}$ the totality of representatives from classes of watermarking equivalence (transversal). So $|J| > O(|K|^{[1/4]})$ because of theorem 2.

We use term *hidden discrete logarithm* for the name of modified algorithm because affine transformation do not allow the adversary to compute the class of watermarking equivalence containing the plaintext (base of the logarithm) and pass in the finite automaton corresponding to the value of the logarithm.

IV. Explicit Construction, Algebraic Graphs of Arithmetical Dynamical System

Missing graph theoretical definitions the reader can find in [1] or [8]. E. Moore [7] used term *tactical configuration* of order (s, t) for biregular bipartite simple graphs with bidegrees $s+1$ and $r+1$. It corresponds to incidence structure with the point set P, line set L and symmetric incidence relation I. Its size can be computed as $|P|(s+1)$ or $|L|(t+1)$.

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

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 *double directed flag graph* on the disjoint union of F_1 with F_2 defined by the following rules

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

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

We will define below the family of graphs $D(k, K)$, where $k > 2$ is positive integer and K is a commutative ring, such graphs have been considered in [5] for the case $K = F_q$.

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 [5] for the case of general commutative ring K:

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

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

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

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

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

(This 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 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 $D(k, K)$.

The incidence relation motivated by the linear interpretation of Lie geometries in terms their Lie algebras. α belongs to the root system

$$\mathrm{Root} = \{(1,0),(0,1),(1,1),(1,2),(2,1),(2,2),(2,2)'\dots, (i,i),(i,i)',(i,i+1),(i+1,i)\dots\}.$$

The "root system"

$$\mathrm{Root} = \{(1,0),(0,1),(1,1),(1,2),(2,1),(2,2), (2,2)'\dots,(i,i),(i,i)',(i,i+1),(i+1,i)\dots\}$$

contains all real and imaginary roots of the Kac-Moody Lie Algebra $\tilde{A}_1$ with the symmetric Cartan matrix. We just doubling imaginary roots (i,i) by introducing $(i,i)'$.

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

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

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 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 the vertex set of the new graph is isomorphic to $K^{n+1} \cup K^{n+1}$. If K is finite, then the cardinality of the colour set is $(|K| - 1)$. Let $\mathrm{Reg}K$ be the totality of regular elements, i.e. not zero divisors. Let us delete all arrows with colour, which is a zero divisor. New graph $RDE(t, K)$ ($RD(K)$) with the induced colouring is the automaton in the alphabet $\mathrm{Reg}(K)$.

Let $P_t(x_{1,0}, x_{0,1}, x_{11}, \dots)$ and $R_t(x_{1,0}, x_{0,1}, x_{11}, \dots)$ are the transition function of infinite graph $RD(K)$ of taking the neighbour of vertex from the first and second copy of the flag set for $D(K)$. The connected components of graph $D(n, K)$ can be given in the following way.

Finally we define the tuple a of theorem 2.

Graph $\phi(n)$ is the double flag graph for $D(k, K)$. We assume that $k \geq 6$and $t = [(n + 2)/4]$. Each flag f from $F_1 \cup F_2$ contains the unique point u $u = (u_{01}, u_{11}, \cdots, u_{tt}, u'_{tt}, u_{t,t+1}, u_{t+1,t}, \cdots)$ of $\mathrm{D}(n, K)$. For every r, $2 \leq r \leq t$, let

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

and $a = a(u) = (a_2, a_3, \cdots, a_t)$. So in fact each polynomial a_i depends really from n variables (see [6]).

V. Time Evaluation

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

The alphabet for password is the same K as for the plaintext. For encryption we use the scheme presented in section (4). The colour of vertex is its first coordinate.

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

All the test were run on computer with parameters:

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

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

A. Comparison our algorithm with RC4

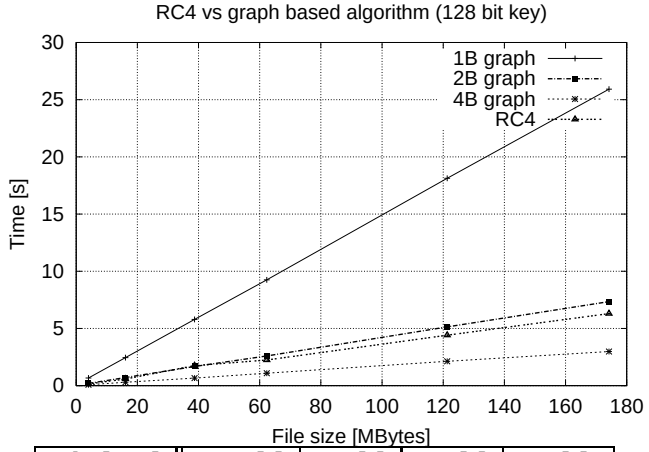
RC4 is a well known and widely used stream cipher algorithm. Protocols SSL (to protect Internet traffic) and WEP (to secure wireless networks) uses it as an option. Nowadays RC4 is not secure enough and not recommended for use in new system. Anyway we chose it for comparison, because of its popularity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

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

TABLE I
TIME GROW FOR $\mathbf{A}_n E_{\bar{a}} \mathbf{A}_n^{-1}$ FOR CHOSEN OPERATOR $\mathbf{A}_n$

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

The mixing properties and speed comparison with DES the reader can find in [4]. The public key algorithms associated with the above dynamical system have been introduced in [11], [13].



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

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

REFERENCES

[1] N. Biggs, *Algebraic Graph Theory* (2nd ed), Cambridge, University Press, 1993.

[2] N. Koblitz, *A Course in Number Theory and Cryptography*, Second Edition, Springer, 1994, 237 p.

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

[4] J. Kotorowicz, V. A. Ustimenko, " On the implementation of cryptoalgorithms based on agebraic graphs over some commutative rings", *Condenced Matters Physics*, Proceedings of the international conferences "Infinite particle systems, Complex systems theory and its application," Kazimerz Dolny, Poland, 2005-2006, 2008, vol. 11, N2(54), 347–360.

[5] 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.

[6] 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.

[7] E. H. Moore,"Tactical Memoranda", *Amer. J. Math.*, v. 18, 1886, 264-303.

[8] R. Ore, *Graph Theory*, London, 1974.

[9] T. Shaska, W. C. Huffman, D. Joener and V. Ustimenko (editors),*Advances in Coding Theory and Cryptography*, Series on Coding Theory and Cryptology, vol. 3, 181-200 (2007).

[10] J. Seberry, J. Pieprzyk, *Cryptography: An Introducion to Computer Security*, Prentice Hall 1989, 379 p.

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

[12] 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 Theory and Cryptology, vol. 3, 181-200 (2007), 131–156.

[13] V. A. 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), 131-186.

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