# On semigroups of multivariate transformations constructed in terms of time dependent linguistic graphs and solutions of Post Quantum Multivariate Cryptography.

Vasyl Ustimenko

University of Maria Curie-Sklodowska, Lublin 20036, Poland

vasyl@hektor.umcs.lublin.pl

**Abstract.**

Time dependent linguistic graphs over abelian group $H$ are introduced. In the case $H=K^*$ such bipartite graph with point set $P=H^n$ can be used for generation of Eulerian transformation of $(K^*)^n$, i.e. the endomorphism of $K[x_1, x_2, \ldots , x_n]$ sending each variable to a monomial term. Subsemigroups of such endomorphisms together with their special homomorphic images are used as platforms of cryptographic protocols of noncommutative cryptography.

The security of these protocol is evaluated via complexity of hard problem of decomposition of Eulerian transformation into the product of known generators of the semigroup. Nowadays the problem is intractable one in the Postquantum setting.

The symbiotic combination of such protocols with special graph based stream ciphers working with plaintext space of kind $K^m$ where $m=n^t$ for arbitrarily chosen parameter $t$ is proposed.

This way we obtained a cryptosystem with encryption/decryption procedure of complexity $m^{1+2/t}$.

**Keywords:** Post Quantum Cryptography, Computer Algebra, time dependent algebraic graphs, affine Cremona semigroup, Eulerian transformations, linguistic graphs over groups and commutative rings.

## 1. Introduction.

Theoretical danger of quantum computers to Information Security has been known since 1994. In the case of asymmetrical cryptography it affects both protocol based cryptosystems for which encryption tools are not given to public and public key cryptosystems.

For instance , a symbiotic combination of Diffie -Hellman protocol (DH) with one time pad encryption or El Gamal cryptosystem in terms of DH algorithm will not be safe in Postquantum era because Discrite logarithm problem is not quantum resistant. Popular RSA public key cryptosystem is not quantum secure because factorisation problem can be solved in polynomial time with the usage of quantum computer.

Nowadays this vulnerability is not only theoretical because large corporations like IBM, Google, governmental scientific centers in Russia and China, UK began to build working quantum computers.

That is why NSA has advised researchers to work on new security products,. NIST and ETSI are currently investigating relevant standards and evaluating proposed public key algorithms. Asymmetrical Cryptography is largely based on theoretical complexity assumptions. Fundamental question whether or not $P=NP$ have been open for decades. This assumption is connected with fundamental conjecture of cryptography that there are no polynomial-time algorithms for solving any $NP$-hard problem.

Such theoretical danger to Cryptography increase interest to problems that are hard to solve in the quantum setting.

Noteworthy that many of the fundamental problems about polynomial time problems have analogs at the level of computability. Many $NP$-hard problems have analogs over various algebraic and combinatorial structures (see [1]). Techniques from computational or statistical algebra afford insights into the distribution of hard instances.

Post Quanntum Cryptography is divided into the following major areas: Code-based cryptography, Lattice-based cryptography**,** Multivariate cryptography. Hash functions base cryptography, Supersingular elliptic curve isogeny cryptography, Noncommutative Cryptography.

This paper is dedicated to a new branch of Multivariate Cryptography (*MC*) dealing with polynomial transformations of affine spaces over various commutative rings of unbounded degree and large semigroups or groups of such transformations. Its content will be presented at ICM -2022 satellite conference “Mathematical aspects of post quantum cryptography”.

Multivariate cryptography is usually defined as the set of cryptographic schemes using the computational hardness of the Polynomial System Solving problem over a finite field. Solving systems of multivariate polynomial equations is proven to be *NP*-hard or *NP*-complete. That is why those schemes are often considered to be good candidates for post-quantum cryptography. Classical Multivariate Cryptography uses systems of quadratic (rarely cubic) equations.
The idea to use degree 2 is motivated by possibility to transform system of equations of any constant degree to equivalent quadratic system of large size. Some weakness of this argument is caused by the fact that such system oftransformations can seriously affect the complexity of system.

The first quadratic multivariate scheme based on multivariate equations was introduced by Matsumoto and Imai in 1988. These authors not only introduced a multivariate scheme but in fact a general principle to design public-key cryptosystems using multivariate equations. There are now plenty of proposals based on this principle that are attractive because they offer the possibility to have very short asymmetric signatures that require only a small amount of resources on embedded devices (see [2]-[6]). After an intense period of cryptanalysis, few schemes emerged as the most robust solutions: HFE (Hidden Field Equations) and UOV (Unbalanced Oil and Vinegar), both developed by J. Patarin in the late 1990s. Variants of these schemes have been submitted to the post-quantum standardization process for public key algorithms as encryption tools or digital signature instruments organized by NIST. For the third round of this competition in July 2020 NIST does not select multivariate algorithm in the category of encryption instruments. Unbalanced Oil and VinegarRemaining Rainbow is investigated as possible digital signature tool.

We believe in the capacity of Multivariate Cryptography (MC) in wide sense as a source of encryption cryptosystems. Recent constructions of families of semigroups and groups of transformation of affine spaces $K^n$ with possibility of computation of $n$ elements from the semigroup in polynomial time gives opportunity to use methods of Semigroup based cryptography in multivariate settings. So instead of one nonlinear transformation of Classical Multivariate Cryptography we can work with several polynomial maps. It allows to construct protocols which security rests on difficult problem to decompose multivariate map
into composition of several given generators. If the map and generators are given in a standard form of Multivariate Cryptography then the decomposition task is intractable problem of Post Quantum Cryptography.
In fact we work in the area of intersection of MC with the Noncommutative Cryptography which uses complexity of problems from Noncommutative algebra on groups, semigroups,algebras and other algebraic systems (see [7]-[15]), recently interesting cryptanalytic results have been obtained in this area [16]-[21].

The output of the protocol can be used for safe creation of multivariate encryption map or safe delivery of such map from one correspondent to another (see [23], [24] and further references, [25], [21], [26]. [27]).This approach can be also used for the construction of digital signatures (see [28], [32]). Note that protocol based
multivariate algorithms essentially differs from traditional for MC public keys. The speed of execution of protocols eseentially differs in the cases of different platforms.

In this paper we continue to use algebraic graphs for the construction of multivariate transformations .We select the case of most efficient *($O(n^3)$)* case of Eulerian transformations [30], [33] and suggest faster algorithm of generation initial data constructed in terms of algebraic graphs theory. This method allows the owner of generating data algorithm a faster restoratation of collision element after receiving data from a partner.

We convert the protocol based on Eulerian transformations of affine space $K^n$ to the cryptosystem of El Gamal type which work with potentially infinite tuples of characters of elements from commutative ring $K$.

In fact, the dimension of plaintext space is $m=O(n^t)$ where parameter $t$ is $>1$.

The symmetric encryption algorithm has complexity $O(m^{1+2/t})$. So, it is possible to work large files. We hope that this postquantum cryptosystem can be used instead of symbiotic combination of classical Diffie-Hellman algorithm and one time pad.

In section 2 we define affine Cremona group and affine Cremona semigroup over general commutative ring $K$ which are central objects of Multivariate Cryptography and Theory of Symbolic Computations. Additionally we

introduce Eulerian semigroup $ES_n(K)$ and group $EG_n(K)$ via endomorphism of $K[x_1, x_2, \dots, x_n]$ moving generic variable $x_i$ into monomial term. We define invertable Jordan-Gauss transformations of $EG_n(K)$ as triangular transformation of $(K^*)^n$ with obvious procedure of reimage computation. Semigroup $EG_n(K)$ satisfies to multiple composition property (MCP), which means ability to compute the composition of $n$ elements in polynomial time.

Other subgroups of $EG_n(K)$ with MCP can be constracted as stable subgroups formed by elements with maximal degree $d$, where $d$ is constant (see [34] and further references).

Third section is dedicated to the concept of linguistic graph over abelian group $G$ which has a imilarity with previously defined linguistic graphs over commutative rings (see [39], [40]). The case $G=K^*$ of commutative ring $K$ is the most important for this paper.

The concept of a time dependent linguistic graph is introduced, this is simply a tuple of linguistic graphs of the same type. We can also consider the time dependent linguistic graphs over commutative .

In section 4 we introduce semigroups ${}^sST_r(K^*)$ of tuples of multivariate maps (elements of Cartesian powers $ES_1(K)$ in the case of $G=K^*$) named as semigroups of symbolic strings.

Effectively computable homomorphism ${}^sST_r(K^*)\rightarrow{}^nES(K)$ of ${}^sST_r(K^*)$ into ${}^nES(K)$ is defined in terms of single time dependent linguistic graph over $K^*$ of type $s, r, m$ where $n=m+s$.

This homomorphism and the concept of Jordan-Gauss transformation allows to define Eulerian transformations with inverting accelerator. It can be used as instrument for the development of public keys algorithms.

Section 5 presents the concept of homomorphisms of time dependent linguistic graphs over $K^*$. Such graph homomorphism induces homomorphism of corresponding subsemigroups of Eulerian transformations. The description of Tahoma protocol in terms of time dependent graph over $K^*$ and its quotients is also presented there.

In section 6 we discuss the usage of pseudorandom sequences or sequences generated by non-deterministic machine for the support of the protocol described in the previous section.

Section 7 gives examples of sparce families of time dependent graphs. They can be used for the faster generation of data for the protocol and faster restoration of collision maps by Alice (creator of protocols data).

The last section contains the description of symbiotic combination of the protocol and graph based stream cipher working with potentially infinite plaintext space.

## 2. On affine Cremona group and Eulerian Transformations.

### 2.1. Formal and affine Cremona groups.

Let $K[x_1, x_2, \dots , x_n]$ be commutive ring of all polynomials in variables $x_1, x_2, \dots , x_n$ defined over a commutive ring $K$. Each endomorphism $F \in E_n(K)$ is uniquely determined by its values on formal generators $x_1$, $i=1,2,\dots, n$. Symbol $End(K[x_1, x_2,\dots , x_n])=E_n(K)$ stands for semigroup of all endomorphisms of $K[x_1, x_2,\dots , x_n]$. So we can identify $F$ and the formal rule $x_1\rightarrow f_1(x_1, x_2,\dots , x_n)$, $x_2\rightarrow f_2(x_1, x_2,\dots , x_n)$, …, $x_n\rightarrow f_n (x_1, x_2,\dots , x_n)$ where $f_i\in K[x_1, x_2,\dots , x_n]$. Element $F$ naturally induces the transformation $\Delta(F)$ of affine space $K^n$ given by the following rule $\Delta(F)$:$(\alpha_1, \alpha_2,\dots, \alpha_n)\rightarrow(f_1 (\alpha_1, \alpha_2,\dots, \alpha_n), f_2(\alpha_1, \alpha_2,\dots, \alpha_n),\dots, f_n(\alpha_1, \alpha_2,\dots, \alpha_n))$ for each $(\alpha_1, \alpha_2,\dots,$

$\alpha_n)\epsilon K^n$. Luigi Cremona (see [22]) introduced $\Delta(E_n(K))= CS(K^n)$ which is currently called *affine Cremona semigroup.* A group of all invertible transformations of $CS(K^n)$ with an inverse from $CS(K^n)$ is known as *affine Cremona group* $CG(K^n)$ (shortly *Cremona group*, see for instance [35], [36]).

We refer to infinite $E_n(K)$ as *formal affine Cremona semigroup*. Density of the map $F$ is the maximal number of monomial terms in $f_i$, $i=1,2,...,n.$

**2.2. Eulerian semigroups.**

Let $K$ be a finite commutative ring with the unit such that multiplicative group $K^*$ of regular elements of this ring contains at least 2 elemments. We take Cartesian power ${}^nE(K) =(K^*)^n$ and consider an Eulerian semigroup ${}^nES(K)$ of transformations of kind

$$
\begin{aligned}
&x_1 \longrightarrow \mu_1 x_1^{\ a(1,1)}\, x_2^{\ a(1,2)} \ldots x_m^{\ a(1,n)},\\
&x_2 \longrightarrow \mu_2 x_1^{\ a(2,1)}\, x_2^{\ a(2,2)} \ldots x_m^{\ a(2,n)}, \qquad\qquad (1)\\
&\ldots\\
&x_m \longrightarrow \mu_n x_1^{\ a(n,1)}\, x_2^{\ a(n,2)} \ldots x_m^{\ a(n,n)},
\end{aligned}
$$

where $a(i,j)$ are elements of arithmetic ring $Z_d$, $d=|K^*|$, $\mu_i \epsilon K^*$.

Let ${}^nEG(K)$ stand for Eulerian group of invertible transformations from ${}^nES(K)$. Simple example of an element from ${}^nEG(K)$ is a written above transformation where $a(i,j)=1$ for $i \neq j$ or $i=j=1,$ and $a(j,j)=2$ for $j \geq 2$. It is easy to see that the group of monomial linear transformations $M_n$ is a subgroup of ${}^nEG(K)$. So semigroup ${}^nES(K)$ is a highly noncommutative algebraic system. Each element from ${}^nES(K)$ can be considered as transformation of a free module $K^n$.

Let $\pi$ and $\delta$ be two permutations on the set $\{1,2,..., n\}$. Let us consider a transformation of $(K^*)^n$, $K=Z_m$ or $K= F_q$ and $d =|K^*|$. We define transformation ${}^AJG(\pi, \delta)$, where $A$ is triangular matrix with positive integer entries $0\leq a(i,j)\leq d$, $i\geq d$ defined by the following closed formula.

$$
\begin{aligned}
&y_{\pi(1)}=\mu_1 x_{\delta(1)}^{\ a(1,1)}\\
&y_{\pi(2)}= \mu_2 x_{\delta(1)}^{\ a(2,1)}\, x_{\delta(2)}^{\ a(2,2)}\\
&\ldots\\
&y_{\pi(n)}= \mu_n x_{\delta(1)}^{\ a(n,1)}\, x_{\delta(2)}^{\ a(n,2)} \ldots x_{\delta(n)}^{\ a(n,n)}
\end{aligned}
$$

where $(a(1,1),d)=1$, $(a(2,2),d)=1$, $\ldots$, $(a(n,n),d)=1.$

We refer to ${}^AJG(\pi, \delta)$ as Jordan - Gauss multiplicative transformation or simply $JG$ element. It is an invertible element of ${}^nES(K)$ with the inverse of kind ${}^BJG(\delta, \pi)$ such that $a(i,i)b(i,i)=1 \ (mod\ d)$. Notice that in the case $K= Z_m$ straightforward process of computation the inverse of $JG$ element is connected to the factorization problem of integer $m$. If $n=1$ and $m$ is a product of two large primes $p$ and $q$ the complexity of the problem is used in RSA public key algorithm. The idea to use composition of $JG$ elements or their generalisations with injective maps *of* $K^n$ into $K^n$ in cryptography was used in [37] ($K=Z_m$) and [38] ($K= F_q$) and [30].

We say that $\tau$ is *a tame Eulerian element* over $Z_m$ or $F_{q.}$ if it is a composition of several Jordan Gauss multiplicative maps over commutative ring or field respectively. It is clear that $\tau$ sends variable $x_i$ to a certain monomial term. The decomposition of $\tau$ into product of Jordan Gauss transformation allows us to find the solution of equations $\tau(x)=b$ for $x$ from $(Z_m^*)^n$ or $(F^*_q)^{\ m}$. So tame Eulerian transformations over $Z_m$ or $F_{q.}$ are special elements of ${}^nEG(Z_m)$ or ${}^nEG(F_q)$ respectively.

We refer to elements of ${}^nES(K)$ as multiplicative Cremona element. Assume that the order of $K$ is a constant. As it follows from the definition the computation of the value of element from ${}^nES(K)$ on the given element of $K^n$ is estimated by $O(n^2)$. The product of two multiplicative Cremona elements can be computed in time $O(n^3)$.

We are not discussing here the complexity of computing the inverse for general element $g\epsilon$ ${}^nEG(K)$ on Turing machine or Quantum computer and the problem of finding the inverse for computationally tame Eulerian elements.

**Remark 2.1.** Let $G$ be a subgroup of ${}^nEG(K)$ generated by Jordan-Gauss elements $g_1, g_{2, \ldots,}$, $g_t$. The *word problem* of finding the decomposition of $g\epsilon G$ into product of generator $g_i$ is a difficult one, i. e. polynomial algorithms to solve it with Turing machine or Quantum Computer are unknown. If the word problem is solved and the inverses of $g_i$ is computable then the inverse of $g$ is determined. Notice that if $n=1$, $K=Z_m$, , $m=pq$ where $p$ and q are large primes and $G$ is generated by $g_1={}_M g_1^a$ the problem is unsolvable by Turing machine but it can be solved with the usage of Quantum Computer.

## 3. Basic constructions.

Similarly to the case of commutative ring (see [39], [40] )we introduce a linguistic graph $I=\Gamma(G)$ over finite abelian group $G$ defined as bipartite graph with a point set $P=P_{s,m}=G^{s+m}$ and a line set $L=L_{r,m}=G^{r+m}$ as linguistic incidence structure $I_m$ if point $x=(x_1, x_2, \dots, x_s, x_{s+1}, x_{s+2}, \dots, x_{s+m})$ is incident to line $y=[y_1, y_2, \dots , y_r , ,y_{r+1}, y_{r+2} , \dots, y_{r+m} ]$ if and only if the following relations hold

$$x_{s+1}^{a(1)} y_{r+1}^{b(1)}=q_1 w_1 ( x_1, x_2 ,\dots , x_s, y_1, y_2, \dots , y_r)$$

$$x_{s+2}^{a(2)} y_{r+2}^{b(2)}=q_2 w_2 ( x_1, x_2 ,\dots , x_s, x_{s+1}, y_1, y_2, \dots , y_r, y_{r+1})$$

$$\dots$$

$$x_{s+m}^{a(m)} y_{r+m}^{b(m)}=q_m w_m( x_1, x_2 ,\dots , x_s, x_{s+1},\dots, x_{s+m-1}, y_1, y_2, \dots , y_r, y_{r+1}, \dots, y_{r+m-1})$$

where $q_j$, $j=1,2,,,m$ are elements of $G$, $w_i$ are words in characters $x_i$ and $y_j$ from $G$ and parameters $a(i)$, $b(i)$ are mutually prime with $d=|G|$. Brackets and parenthesis allow us to distinguish points from lines similarly to the case of linguistic graphs over commutative rings.

We define *colours* $\rho((p))$ and $\rho([l])$ of the point $(p)$ and the line $[l]$ as the tuple of their first coordinates of kind $\boldsymbol{a}=(p_1, p_2, \dots, p_s)$ or $\boldsymbol{a}=(l_1 , l_2 , \dots, l_r )$ and introduce well defined operator $N(v, a)$ of computing the neighbour of vertex $v$ of colour $a \epsilon G^s$or $a \epsilon G^r$. Similarly to the case of linguistic graph over commutative *ring* we define *colour jump* operator $J(p, a)$, $a \epsilon G^s$ on partition set $P$ and $J(l,a)$, $a \epsilon G^r$ on partion set $L$ by conditions $J(p,a)=(a_1, a_2, \dots a_s, p'_{1+s}, p/_{2+s}, \dots, p'_{s+n})$ and $J(l,a))=[a_1, a_2, \dots a_r, l_{1+r}, l'_{2+r}, \dots, l'_{r+m}]$.

*If* $G'>G$ then we can consider graph $I(G')$ of type $(r, s, m)$ with partition sets $P'=(G')^{m+s}$ and $L'=(G')^{m+r}$ given by the same equations with $q_i$ from $G$. Note that group $G'$ can be infinite one.

Let $x_1, x_2, \dots, x_n$ be the list of variables. We define $G<x_1, x_2, \dots, x_s>$ as a totality of monomial terms with coefficients from $G$ of kind $gx_1^{a(1)}x_2^{a(2)}\dots x_n^{a(n)}$, where $a(i)$, $i=1,2,\dots,n$ are elements of $Z_d$, $d=|G|$.

We introduce ${}^sB_s(G)$ as $(G<x_1, x_2, \dots, x_s>)^s$ and ${}^rB_s$ as $G(<x_1, x_2, \dots, x_s>)^r$. Element $(f_1, f_2, \dots, f_s)$ from ${}^sB_s(G)$ can be identified with the endomorphism $x_1 \rightarrow f_1, x_2 \rightarrow f_2, \dots, x_s \rightarrow f_s$ of $G<x_1, x_2, \dots, x_s>$ as a group with the operation given via the following ring $gx_1^{a(1)}x_2^{a(2)}\dots x_n^{a(n)} \circ g' x_1^{a(1)'} x_2^{a(2)'}\dots x_n^{a(n)'}=gg' x_1^{a(1)+a(1)'} x_2^{a(2)+a(2)'}\dots x_n^{a(n)+'(n)'}$.

We assume that $G=K^*$ for the commutative ring $K$.

We consider ${}^sBS_r(K^*)=K^*<x_1, x_2, \dots, x_s>^r$. Element $H=(g(1)\ x_1^{a(1,1)}x_2^{a(1,2)}\dots x_s^{a(1,s)}, g(2)\ x_1^{a(2,1)}x_2^{a(2,2)}\dots x_s^{a(s,s)}, \dots, g(s)\ x_1^{a(s,1)}x_2^{a(s,2)}\dots x_s^{a(s,s)})$ from ${}^sBS_s(K^*)$ can be identified with endomorphism $H: x_1 \rightarrow g(1)x_1^{a(1,1)}x_2^{a(1,2)}\dots x_s^{a(1,s)}$, $x_2 \rightarrow g(2)x_1^{a(2,1)}x_2^{a(2,2)}\dots x_s^{a(s,s)}, \dots$, $x_s \rightarrow g(s)\ x_1^{a(s,1)}x_2^{a(s,2)}\dots x_s^{a(s,s)}$ of $K^*<x_1, x_2, \dots, x_s>$ from $ES_s(K)$.

Endomorphism $H$ acts naturally on ${}^sBS_r(K^*)$. The result of action of $H$ on $G$ from ${}^sBS_r(K^*)$ will be written as $G(H)$ or simply $GH$.

We consider the concept of time dependent linguistic graph over commutative group $K^*$.

Let $L_{r,s,m}(K^*)=L(K)$ be variety of all linguistic graphs of type $(r,s, m)$ over $K^*$. $F(L_{r,s,m}(K))=F(L(K))$ stands for the free semigroup over the alphabet $L(K)$.

We interpret word $(I(1), I(2), \dots, I(l))=F(L)$ as time dependend linguistic graph $I_t (K^*)$ on interval $[1, l]$ and refer to $j$ of $I(j)$ as time parameter. We think that there is given by some Oracle function, which establishes coefficients $q_i =q_i (t)$ from $K^*$, $a(i,j)=a(i,j)(t)$ from $Z_d$ , $i=1,2,\dots,m$,
$j=1,2,\dots,m$. Product of $(I(1), I(2), \dots, I(l))$ with $(I'(1), I'(2), \dots, I'(p))$ is time dependent graph (shortly t.d.g.) $(I(1), I(2), \dots, I(l), I(l+1), I'(l+2), \dots, I'(l+p))$.

With time dependent graph $I_t$ we associate transformation on the point set $K^{s+m}$ (or line set $K^{r+m}$) from group $ES_{s+m}(K)$ ( $ES_{r+m}(K)$ respectively) via the following constructions.

**4. Semigroups related to time dependent linguistic graphs.**

Let us consider special subsemigroup ${}^sST_r(K^*)$ and its special homomorphisms into Eulerian semigroup ${}^nES(K)$, $n=m$.

We consider totality of tuples of kind $u=(H_1, G_1, G_2, H_2. H_3, G_3, G_4, H_4, \dots, H_{2t-1}, G_{2t-1}, G_{2t}, H_{2t}, H_0)$ where $H_i$ and $G_i$ are elements of ${}^sB_s(G)$ and ${}^sB_r(G)$ respectively. Assume that ${}^sST_r(K^*)$ is a totality of such tuples of various length of kind $4t+1$where $t=0, 1, 2, \dots$. We refer to these tuples as symbolic strings of typer $s$, $r$.

Let us select time dependent linguistic graph $I=I_t$ from $L_{s,r,m}(K^*)$, defined on the interval $[1,t]$. Let us expand $K^*$ for $R=K^*<x_1, x_2, \dots, x_{s+m}>$. This change instantly converts t.d.g. $I_t=(I(1), I(2), \dots, I(t))$ into $I'_t=(I'(1), I'(2), \dots, I'(t))$ from $F(L_{s,r,m}(R))$.

*Time dependent walk with colour jumps* in the $I'_t(R)$ starts in the graph $I'(1)$ with selection of *initial point* $v_0=(x_1, x_2, \dots, x_{s+m},)$
from $R^{s+m}$. Further construction of the walk is prescribed by symbolic string $u$.

During initial time interval $(0, 1]$ we have to compute
$J(v_0, H_1)=v_1$, $N(v_1, G_1)=v_2$, $J(v_2, G_2)=v_3$ and $N(v_3, H_2)=v_4$ in the graph $I'(1)$. Doing these computations we use only multiplication of $K^*<x_1, x_2, \dots, x_{s+m}>$.

Next step corresponds to time interval $(1, 2]$. We treat the output $v_4$ of computations within interval $(0,1]$ as point of the graph $I'(2)$. Now we compute $J(v_4, H_3)=v_5$, $N(v_5, G_3)=v_6$, $J(v_6, G_4)=v_7$ and $N(v_7, H_2)=v_8$ in the graph $I'(2)$.

Continuation of this process partited into $t$ steps to the klast four vertices of graph $I'(t)$ given by the list
$J(v_{4t-4}, H_{2t-1})=v_{4t-3}$, $N(v_{4t-3}, G_{2t-1})=v_{4t-2}$, $J(v_{4t-2}, G_{2t})=v_{4t-1}$ and $N(v_{4t-1}, H_{2t})=v_{4t.}$

Finally we compute $v_{4t+1}=J(v_{4t}, H_0)$ from ${}^sB_s(R)$ in the graph $I'(t)$.

Noteworthy that output $v_{4t+1}$ *is a* tuple $({}^1H_0, {}^2H_0, \dots, {}^sH_0,$
$F_{s+1}, F_{s+2}, \dots, F_{s+m})$ where $H_0=({}^1H_0, {}^2H_0, \dots, {}^sH_0)$,
$F_j$ are elements of $K^*<x_1, x_2,, \dots, x_s>$.

The output defines the map ${}^I\acute{\eta}$:
$x_1 \to {}^1H_0, x_2 \to {}^2H_0, \dots, x_s \to {}^sH_0,, x_{s+1} \to F_{s+1}, x_{s+2} \to F_{s+2}, \dots, x_{s+m} \to F_{s+m}$ from $E_{s+m}(K)$.

Above presented algorithm defines map from ${}^sST_r(K^*)$ into ${}^nES(K)$, $n=m$ induced by the time dependent graph. Let us concider these maps in formal way.

We define a product of $u=(H_1, G_1, G_2, H_2. H_3, G_3, G_4, H_4, \dots, H_{2t-1}, G_{2t-1}, G_{2t}, H_{2t}, H_0)$ and $v=(H'_1, G'_1, G'_2, H'_2. H'_3, G'_3, G'_4, H'_4, \dots, H'_{2k-1}, G'_{2k-1}, G'_{2k}, H'_{2k}, H'_0)$ *as* $u \circ v=(H_1, G_1, G_2, H_2. H_3, G_3, G_4, H_4, \dots, H_{2t-1}, G_{2t-} H_{2t}, H'_1H_0, G'_1H_0, G'_2H_0, H'_2H_0, H'_3H_0, G'_3H_0, G'_4H_0, H'_4H_0, \dots, H'_{2k-1}H_0, G'_{2k-1}H_0, G'_{2k},H_0, H'_{2k}H_0, H'_0H_0)$.

It is easy to see that this products converts ${}^sST_r(K^*)$into a semigroup.

The totality of symbolic strings of length $1$ forms subsemigroup isoporphic to ${}^sES(K)$. The unity of ${}^sST_r(K^*)$ is the unity of *semigroup of symbolic strings* ${}^sST_r(K^*)$.

**Lemma 1.** *Let $I=I_t$ be time dependent linguistic graph of type $(s, r, m)$ of length $t$. The map ${}^I\acute{\eta}$: ${}^sST_r(K^*) \to {}^nES(K)$ is a homomorphism of semigroups.*

We refer to ${}^I\psi({}^sST_r(K^*))={}^IS(K^*)$ as a chain transitions semigroup of linguistic graph $I(K^*)$ over $K^*$ and to map ${}^I\acute{\eta}$ as multiplicative linguistic compression map of $I_t$.

We consider a direcr product of ${}^sST_r(K^*)$ and $F(L_{r,s,m}(K))$ and homomorphism $\acute{\eta}={}^m\acute{\eta}$ of ${}^sST_r(K^*) \circ F(L_{r,s,m}(K))$ defined by the rule
$(u, I) \to {}^I\acute{\eta}(u)$.

**Lemma 2.** *The map $\acute{\eta}$ is the homomorphism of ${}^sST_r(K^*) \circ F(L_{r,s,m}(K))$ into ${}^nES(K)$, $n=s+m$.*

We refer to ${}^sST_r(K^*) \circ F(L_{r,s,m}(K))$as *space of symbolic walks of type $(s, r)$* and say that $\acute{\eta}$ is a *compression map.*

We refer *to* $\acute{\eta}({}^{s}ST_r(K^*)\circ F(L_{r,s,m}(K))= {}^{s,m}S_r(K^*)$ as *chain transition subsemigroup of* ${}^{n}ES(K)$, $n=s+m$.

Let $(u, I_t)$ be a symbolic walk from ${}^{s}ST_r(K^*)\circ F(L_{r,s,m}(K))$.

We refer to $\acute{\eta}(u, I_t)$ as *chain transition of time dependent linguistic graph* $I_t$.

Let ${}^{s}GT_r(K^*)$ be subsemigroup of ${}^{s}GT_r(K^*)$ formed by symbolic strings of kind $u=(H_1, G_1, G_2, H_2. H_3, G_3, G_4, H_4,..., H_{2t-1}, G_{2t-1}, G_{2t}, H_{2t}, H_0)$ where $H_0$ is an element of $EG_s(K)$

**Lemma 3.** *The map* $\acute{\eta}$ *induces homomorphism* $\acute{\eta}'$ *of* ${}^{s}GT_r(K^*)\circ F(L_{r,s,m}(K))$ *into* ${}^{k}EG(K)$, $k=s+m$.

We refer to $\acute{\eta}'({}^{s}GT_r(K^*)\circ F(L_{r,s,m}(K)) = {}^{s,m}G_r(K^*)$ as *chain transition group of type (s, r, m).*

Assume that $\acute{\eta}'(u, I_t)=F$ is written in its standard form, $s=O(1)$ and $r=O(1)$. The knowledge of some reimage of ή' and $H'_0=(H_0)^{-1}$ allow us to compute $F^{-1}(y)$ *in given* $y$ in time $O(tn^2)$.

Really we can form the reverse string $v=(H_{2t}H'_0, G_{2t}H'_0, G_{2t-1}H'_0, H_{2t-1}H'_0. H_{2t-2}H'_0, G_{2t-2}H'_0, G_{2t-3}H'_0, H_{2t-3}H'_0,..., H_2H'_0, G_2H'_0, G_1H'_0, H_1H'_0, H'_0)$ and check that $\acute{\eta}'(uv)$ is an identity map.

Let us consider rhe data $D$ consisting of symbolic walk $(u, I_t)$, where $I_t$ is time dependent graph of type $(s,r, m)$, element $u$ of ${}^{s}GT_r(K^*)$ and two lists of Jordan - Gauss generators of $EG_{s+m}(K)$ formed by $G_1, G_2,..., G_{t(1)}$ and $F_1, F_2,..., F_{r(2)}$ with $t(1)\geq 1$, $t(2)\geq 1$.

We say that element $F= G_1G_2...G_{t(1)}\acute{\eta}(u, I_t)F_1F_2,...F_{r(2)}$ written in its standard form has *inverting accelerator D.*

Noteworthy that knowledge of $D$ allow us to find $F^{-1}$ in its standard *form* in polynomian time in variable $k=m+s$.

Pairs of kind $(F, D)$ can be used instead of products of Jordan Gauss elements for the constructions of public key cryptosystems introduced in [37], [38].

## 5. Quotients of time dependent graphs and protocols of Noncommutative Cryptography.

Let us consider time dependent linguiticn graph $I_t=(I(1), I(2),...,I(t))$ over group $K^*$ of type $(r,s,m)$ and parameter $n$, $n<m$.

We can define another time dependent linguistic grap $\mu_n(I_t)$ = $(I'(1), I'(2),..., I'(t))$ where $I'(l)$ is obtained from $I(t)$ by deleting of last coordimnates $x_{n+s+1}. x_{n+s+2}, ..., x_{m+s}$ of points and $\mathbf{y}_{n+s+1}. y_{n+s+2}, ..., y_{m+s}$ of lines and cancellation of last $n$-$m$ equations in the definition of $I_t$ **.**The map $\mu_n$ is the homomorphism of semigroups $F(L_{r,s,m}(K))$ onto $F(L_{r,s,n}(K))$.

It induces the homomorphism $\pi_n$ of ${}^{s}ST_r(K^*)\circ F(L_{r,s,m}(K))$ onto ${}^{s}ST_r(K^*)\circ F(L_{r,s,n}(K))$ acting by the rule $\pi_n(u, I_t)=(u, \mu_n(I_t))$. Composition of $\pi_n$ and ${}^{n}\acute{\eta}$ is a homomorphism of ${}^{s}ST_r(K^*)r\circ F(L_{r,s,m}(K))$ onto ${}^{s,n}S_r(K^*)$.

In fact the following diagram is commutative

$$\begin{tikzcd}
{}^{s}ST_r(K^*)\circ F(L_{r,s,m}(K)) \arrow[r, "{}^{m}\acute{\eta}"] \arrow[d, "\mu_n"] & {}^{s,m}S_r(K^*) \arrow[d, "\tau_n"] \\
{}^{s}ST_r(K^*)\circ F(L_{r,s,n}(K)) \arrow[r, "{}^{n}\acute{\eta}"] & {}^{s,n}S_r(K^*)
\end{tikzcd}$$

where $\tau_n$ is the restriction of endomorphism of $K[x_1.x_2,...,x_{m+s}]$ on $K[x_1.x_2,...,x_{n+s}]$ given by its values on $x_1.x_2,...,x_{n+s}$,.

Below we consider Tahoma protocol for platforms defined by linguistic graphs. Noteworthy that word *Tahoma* is the abreviation of tame homomorphism.

PROTOCOL.

Alice takes elements $c_1, c_2, ..., c_{k(1)}$, $k(1)\geq 2$, $k(1)=O(1)$ from ${}^{s}ST_r(K^*)\circ F(L_{r,s,m}(K))$ of length $t(i)\geq 2$, $i=1,2,...,k(1)$ and $d_1 =(u, I_t)$, $I_t=(I(1),I(2),...,I(t))$ is an element from ${}^{s}GT_r(K^*)\circ F(L_{r,s,m}(K))$, $u=(H_1, G_1, G_2, H_2,\ldots,H_{4t-1}, G_{4t-1}, G_{24t-1}, H_{4t}, H_0)$. She computes $d'_1=(rev(u), (u(t), u(t-1),...,u(1))$. Alice computes $d_1c_1d'_1, d_1c_2d'_1, ..., d_1c_{k(1)}d'_1$ in the group ${}^{s}ST_r(K^*)\circ F(L_{r,s,m}(K))$. She applies ${}^{m}\acute{\eta}$ to these elements and gets $z_1= {}^{m}\acute{\eta}(d_1c_1d'_1)$, $z_2= {}^{m}\acute{\eta}(d_1c_2d'_1)$, ..., $z_{k(1)}= {}^{m}\acute{\eta}(d_1c_2d'_1)$.

Alice takes some Jordan–Gauss generators $J_1, J_2,..., J_{k(2)}$, $k(2)\geq 1$,

from $EG_{m+s}(K)$ and computes their inverses $J'_1, J'_2, \ldots, J'_{k(2)}$ and
$J= J_1J_2\ldots, J_{k(2)}$ with $J^{-1}$. She forms $a_1 = J z_1 J^{-1}$, $a_2 = J z_2 J^{-1}, \ldots, a_{k(1)} = J z_{k(1)} J^{-1}$

Alice computes $c'_1=\mu_n(c_1), c'_2=\mu_n(c_2), \ldots, c'_{k(1)}=\mu_n(c_{k(1)})$, She takes $d_2 =(v, I'_{t'})\epsilon\, {}^sGT_r(K^*)\circ F(L_{r,s,n}(K))$,, where $I'_{t'}$ has type $(s, r, n)$ ,
$v=(H'_1, G'_1, G'_2, H'_2, \ldots, H'_{4t'-1}, G'_{4t'-1}, G'_{4t'}, H'_{4t'}, H'_0)$ and forms $rev(d_2)=d'_2$. Alice constructs $y_1= {}^n\acute{\eta}( d_2c'_1d'_2 )$ , $y_2= {}^n\acute{\eta}(d_2c'_2d'_1), \ldots, y_{k(1)}= {}^n\acute{\eta}(d_2c'_{k(1)}d'_2)$.
She takes some Jordan – Gauss generators $G_1, G_2, \ldots, G_{k(3)}$, $k(3) \geq 1$
from $EG_{n+s}(K)$ and computes their inverses $G'_1, G'_2, \ldots, G'_{k(3)}$ and
$G= G_1G_2\ldots, G_{k(3)}$ with $G^{-1}$.

Alice forms $b_1 = Gy_1G^{-1}$, $a_2 = G y_2 G^{-1}, \ldots, b_{k(1)} = Gz_{k(1)}G^{-1}$

She sends pairs $(a_i, b_i)$, $i=1,2,\ldots, k(1)$ to Bob.
Bob takes tuple $(j((1), j(2),\ldots,j(q))$, $q=O(1)$, $q>1$ where $j(i)\epsilon\{1,2,\ldots,k(1)\}$ such that $|\{ j((1), j(2),\ldots,j(q)\}|\geq 2$. He forms $a=a_{1j(1)}\ a_{1j(2)\ldots,}\ a_{j(q)}$ and sends it to Alice. Bob computes $b=b_{j(1)}b_{j(2)\ldots,}b_{j(q)}$ and keeps it safely in his private storage.

Alice computes ${}^1a=J^{-1}aJ$, ${}^2a={}^m\acute{\eta}(rev(d_1))^1a^m\acute{\eta}(d_1)$,
$\tau_n({}^2a)= {}^1b$, ${}^2b = {}^n\acute{\eta}({}^2b)$ and collision element $b$ as $G({}^2b)G^{-1}$. Note that $b$is an element $ES_{n+s}(K^*)$.

**Remark 1.** The security of protocol rests on the complexity of problem of decomposition of element a from $ES_n(K)$ in the composition of generators $a_i$, $i=1,2,\ldots, k(1)$.

This problem is intractable even in the case of usage Turing machine jointly with Quantum computer,

### 5. On the usage of pseudorandom or genuinely random sequences.

Let us consider the algorithm of data generation for the protocol.
Single linguistic graph over $K^*$ of type $(s. r, m)$ can be written as

$$x_{s+1}^{a(1)} y_{r+1}^{b(1)}=q_1x_1^{a(1,1)}x_2^{a(1,2)} \ldots x_s^{a(1,s)}\, y_1^{b(1,1)},y_2^{b(1,2)}\ldots\ y_r^{b(1,r)}$$

$$x_{s+2}^{a(2)}y_{r+2}^{b(2)}=q_2x_1^{a(2,1)}x_2^{a(2,2)} \ldots x_s^{a(2,s)}x_{s+1}^{a(2,s+1)}y_1^{b(2,1)}y_2^{b(2,2)}\ldots y_r^{b(2,r)}$$
$$y_{r+1}^{b(2,r+1)}$$

$$\ldots$$

$$x_{s+m}^{a(m)}y_{r+m}^{b(m)}=q_mx_1^{a(m,1)}x_2^{a(m,2)}\ldots x_s^{a(m,s)}x_{s+1}^{a(m,s+1)}\ldots, x_{s+m-1}^{a(m,s+m-1)}$$

$$y_1^{b(m,1)}y_2^{b(m,2)} \ldots y_r^{b(m,r)}y_{r+1}^{b(m,r+1)}\ldots y_{r+m-1}^{b(m,r+m-1)}$$

For the protocol Alice needs $t(1)+t(2)+\ldots+t(k(1))+t=T$ such graphs.
So she need the tuple from $(Z^*_d)^{2mT}$ to form lefthandside of graphs equations.
Additionally she need the tuple from $(K^*)^{mT}$ and the tuple from
$Z_d^{(m(s+r)+m(m-1))T}$.
Another tuple from $K^{nt'}Z^{*\,2nt'}_d Z_d^{T(n(s+r)+n(n-1))t'}$ defines time dependent graph $I'_{t'}$.

We assume that Jordan-Gauss transformation $J$ from $ES_k(K)$, $k=m+s$ has form $J_1J_2$ where $J_1$ is given by
$x_1\rightarrow q(1)x_1^{c(1,1)}, x_2\rightarrow q(2)x_1^{c(2,1)}x_2^{c(2,1)}\quad \ldots x_n\rightarrow q(k)x_1^{c(k,1)}x_2^{c(k,2),}\ldots x_n^{c(k,k)}$ with elements $c(1,1), c(2,2),\ldots., c(k,k)$ from $Z^*_d$ and $J_2$ is given by $x_1\rightarrow p(1)x_1^{e(1,1)}x_2^{c(1,2)}\ldots x_n^{c(1,k)}$, $x_2\rightarrow p(2)x_1^{e(2,1)}$ $x_2^{e(2,2)}\ldots x_{n-1}^{e(2,k-1)}$, $\ldots$,
$x_n\rightarrow p(n)x_1^{e(k,1)}$ with
elements $e(1,n), e(2,n-1),\ldots,e(k,1)$ from $(Z^*)^d$.

So variety of pairs $J_1, J_2$ is isomorphic to $(Z^*_d)^{2k}(K^*)^{2k}Z_d^{k(k-1)}$.
Last coordinates $H_0$ and $H'_0$ can be also constructed as products of lower and upper triangular $JG$ elements. So pair $(H_0, H'_0)$ corresponds to element from $(Z^*_d)^{4s}(K^*)^{4s}Z_d^{2s(s-1)}$. Other components of $u$ and $v$ together corresponds to the tuple from $(K^*)^{2(s+r)(t+t')}Z_d^{2s(s+r)(t+t')}$.

Additionally Alice need elements z of ${}^sST_r(K^*)$ of length $t_1, t_2, \ldots t_k(1)=T'$ together they correspond to element from $(K^*)^{2(s+r)T'}Z_d^{2s(s+r)T'}$.

Alice uses pdeudorandom generations of sequences in the alphabets $K^*$, $Z^*_d$, $Z_d$ and form entrance data for the protocol.

**Remark 2.** Complexity of the algorithm coincides with the complexity $O(m^3)$ of the composition of two elements from $ES_{m+s}$.
**Remark 3.** Alice can speed up her part of computations. She can compute data $(a_i, b_i)$, $i=1,2,\dots, k(1)$, $k(1)=O(1)$ via usage of sparce linguistc graphs over $K^*$ with $O(n)$ nonzero entries from $Z_d$ and sparce Eulerian transformations $J$ and $G$. For sparce graph computation of the neighbour of vertex with selected colour costs $O(n)$. In this case preparation of the data costs Alice $O(n)$ elementary operations. She can decrypt in time $O(n^2)$.

## 6. Examples of sparce graphs over *K** and commutative ring *K*.

Recall that linguistic graph $I=\Gamma(K)$ over finite commutative ring K is defined as bipartite graph with a point set $P=P_{s,m}=K^{s+m}$ and a line set $L=L_{r,m}=K^{r+m}$ of incidence structure $I_m$ such that point $x=(x_1, x_2, \dots, x_s, x_{s+1}, x_{s+2}, \dots, x_{s+m})$ is incident to line $y=[y_1, y_2, \dots, y_r, y_{r+1}, y_{r+2}, \dots, y_{r+m}]$ if and only if the following relations hold

$$a(1)x_{s+1} + b(1)y_{r+1}=f_1(x_1, x_2, \dots, x_s, y_1, y_2, \dots, y_r)$$

$$a(2)x_{s+2} + b(2)y_{r+2}=f_2(x_1, x_2, \dots, x_s, x_{s+1}, y_1, y_2, \dots, y_r, y_{r+1})$$

$$\dots$$

$$a(m)x_{s+m} + b(m)y_{r+m}=f_m(x_1, x_2, \dots, x_s, x_{s+1}, \dots, x_{s+m-1}, y_1, y_2, \dots, y_r, y_{r+1}, \dots, y_{r+m-1})$$

where $a(j)$ and $b(j)$, $j=1,2,,,m$ are elements of $K^*$, $f_i$ are polynomials from variables $x_i$ and $y_j$ written in brackets. Brackets and parenthesis allow us to distinguish points from lines (see [40]).

We define *colours* $\rho((p))$ and $\rho([l])$ of the point $(p)$ and the line $[l]$ as the tuple of their first coordinates of kind $\boldsymbol{a}=(p_1, p_2, \dots, p_s)$ or $\boldsymbol{a}=(l_1, l_2, \dots, l_r)$ and introduce well defined operator $N(v, a)$ of computing the neighbour of vertex $v$ of colour $a\epsilon K^s$ or $a\epsilon K$. Similarly to the case of linguistic graph over commutative ring we define recolouring operator $J(p, a)$, $a\epsilon K^s$ on partition set $P$ and $J(l,a)$, $a\epsilon K^r$ on partion set $L$ by conditions $J(p,a)=(a_1, a_2, \dots a_s, p'_{1+s}, p/_{2+s}, \dots, p'_{s+n})$ and $J(l,a))=[a_1, a_2, \dots a_r, l_{1+r}, l'_{2+r}, \dots, l'_{r+m}]$.

Assume that well known graph $A(k, K)$ (see [29] or [42], [43]) of type $(1,1,n-1)$ is given by equations $x_2-y_2=y_1x_1$, $x_3-y_3=x_1y_2, \dots, x_k-y_k=y_1x_{k-1}$ in the case of even $k$.

We consider linguistic graph $A(k, K^*)$ over commutative group $K^*$ *of* type $(1, 1, k-1)$ given by equations $x_2/y_2=y_1x_1$, $x_3/y_3=x_1y_2, \dots, x_k/y_k=y_1x_{k-1}.$

We define class of time dependent graphs ${}^DA_t(k, K^*)$ given by equations $x_2^{a(1,t)}y_2^{b(1,t)}=y_1^{c(1,t)}x_1^{d(1,t)}$, $x_3^{a(2,t)}y_3^{b(2,t)}=x_1^{c(2,t)}y_2^{d(2,t)}, \dots,$ $x_k^{a(k-1,t)}y_k^{b(k-1,t)}=y_1^{c(k-1,t)}x_{k-1}^{d(k-,t)}$ with $a(i,t)$, $b(i,t)$, $c(i,t)$, $d(i,t)$ from $Z_d$-$\{0\}$, $d=|K^*|$, $i=1,2,\dots,k-1$, $t=1,2,\dots,T$ such that $a(i)$ and $b(i)$ are mutually prime with $d$. The graph depends on data $D$ given by array $(a(t), b(t), c(t), d(t))$, $t\epsilon[1, T]$.

These graphs can be used for the implementation of the protocol together with tansformations of kind $J$: $x_1 \rightarrow qx_1^{m(1)}x_2^{m(2)} \dots x_k^{m(k)}$, $x_i \rightarrow x_i$, $i=2,3,\dots,k.$

## 6, Expansions of the protocol to cryptosystems.

Let us assume that collision element $b$ is written in its standard form

$$x_1 \rightarrow q_1x_1^{a(1,1)}x_2^{a(1,2)} \dots x_m^{a,(1,k)},$$
$$x_2 \rightarrow q_2x_1^{a(2,1)}x_2^{a(2,2)} \dots x_m^{a(2,k)}, \qquad (1)$$
$$\dots$$
$$x_k \rightarrow q_kx_1^{a(k,1)}x_2^{a(k,2)} \dots x_m^{a(k,k)},$$

where $k=n+s$. We assume that parameter $k$ is even.

We are going to expand the ptotocol to cryptosystems via multiple usage of cubical multivariate encryption maps based on graphs $A(n, K)$ with large parameter $t=O(k^a)$, $\alpha>1$.

For this purpose we form $b(i,j)=(q_iq_j)^{a(i,j)}$ and transformation $T_i$ of kind $x_i \rightarrow b(i,1)x_1 + b(i,2)x_2 \dots + b(i,k)x_k + x_{k+1} + x_{k+2} \dots + x_t$, $x_j \rightarrow x_j$, $j=2,3,\dots,t$. We assume that $T_{k+1}: x_1 \rightarrow q_1x_1 + q_2x_2 \dots + q_kx_k + x_{k+1} + x_{k+2} \dots + x_t$. ,$x_j \rightarrow x_j$, $j=2,3,\dots,t$.

Let $\gamma=(b(1),b(2),,\dots,b(k))$ be tuple of elements of $K$-$\{0\}$.

We define the transformation ${}^{\gamma}N$ via the following recurrent algorithm

Consider graph $A(t,K[x_1, x_2, \dots, x_t])$ and
start path with the initial point $v_0=(x_1, x_2,\dots, x_t)$ and consequtive elements $v_1=N(v_0, x_1+b(1))$, $v_2=N(v_1, x_1+b(2))$,$v_3=N(v_2, x_1+b(1)+b(3))$, $v_4=N(v_3, x_1+b(2)+b(4))$,…,
$v_{k-1}=N(v_{k-2},x_1+b(1)+b(3)+\dots+b(k-1))$,$v_k=N(v_{k-1}, x_1+b(2)+b(4)+\dots+b(k))=(u(1), u(2), \dots, u(t))$. Finally we define
${}^{\gamma}N=N_{b(1),b(2),b(1)+b(3), b(2)+b(4),\dots b(1)+b(3)+\dots b(k-1),b(2)+b(4)\dots,+b(k)}$ map from $EG_t(K)$ sending $x_i$ to $u_i \in K[x_1, x_2,\dots, x_t]$.

Inverse map for ${}^{\gamma}N$ *is* ${}^{\gamma}N'=N_{-b(k)-b(k-2)-\dots-b(2)+b(1)+b(3)+\dots+b(k-1),-b(k), \quad \dots ,-b(k)-b(k-2)-\dots-b(4) ,-b(k)-b(k-2)-\dots-b(2)+b(1) ,-b(k)-b(k-2)-\dots-b(2)}$ defined by path $w_0=(x_1, x_2,\dots, x_t)$, $w_1=N(w_0, ,x_1-b(k)-b(k-2)-b(2)+b(1)+b(3)+b(k-1))$, $w_2=N(w_1, x_1-b(k))$, $w_3=N(w_2, x_1-$
$-b(k)-b(k-2)-b(2)+b(1)+b(3)+\dots+b(k-3)$, $w_4=N(w_1, x_1-b(k)-b(k-2)$,…
$N(w_{k-1}, x_1-b(k)-b(k-2)-\dots-b(2)+b(1)$, $N(w_k, x_1-b(k)-b(k-2)-\dots-b(2))$

We know that in the case of $\alpha>1$ and fixed tuple $a=(a_1, a_2,\dots, a_t)$ for distinct $\gamma$ and $\gamma'$ we have ${}^{\gamma}N(a)\neq{}^{\gamma'}N(a)$ (see [42]).

Let us consider columns $\gamma(i)=(b(i,1), b(i,2), \dots,b(i,k))$ of matrix $B=(b(i,j))$ and corresponding cubical maps ${}^{\gamma(i)}N=N_i$ of $K^t$ to $K^t$.

**The cryptosystem.**

Correspondents execute protocol. They us its output to form matrix $B$ and select the parameter $t$, $t=O(n^{\alpha})$, $\alpha>1$ for work with cipherspace $K^t$.

Alice and Bob compute $T_i$, $i=1,2,\dots, k+1$ and $N_i$ .

They use encryption map $E=T_1N_1T_2N_2\dots T_kN_kT_{k+1}$ . They do not need compute the standard form of $E$. Knowledge of the decomposition of $E$ allow them to encrypt in time $O(kt)=O(t^{1+1/\alpha})$. In the case of large parameter α the execution spead is close to procedure of reading file $O(t)$, The speed of decryption procedure takes the same type.

Properties of stable cubical transformations $N_i$ and unstable cubical transformation $T_jN_i$ and $T_j N_i T_l$ are well unvestigated.

For instance densities of $T_jN_iT_l$, i.e. total number of monomials in their stzndard forms is at least $O(tk^2)$.

So, theoretically map $E$ has exponential degree $O(3^k)$ and density $O(t2^k)$.

So, linearisation attacks of adversary or usage of other cryptanalytic tools are unfeasible.

Adversary has to break the protocol via the solution of word decomposition problem in a semigroup $ES_k(K)$.

## References

[1] L. Babai, Graph Isomorphism in Quasipolinomial Time}, arXive: 1512 03547v1 [cs. DS], 11 Dec 2015.

[2] N. Koblits, Algebraic Cryptography, Springer, 2000.

[3] J. Ding, J. E. Gower, D. S. Schmidt, Multivariate Public Key Cryptosystems}. Springer, Advances in Information Security, V.~25, 2006.

[4] Jacques Patarin, Louis Goubin, Trapdoor one-way permutations and multivariate polynominals, ICICS 1997, 356-368.

[5] Jacques Patarin, Louis Goubin, Asymmetric cryptography with S-Boxes, ICICS 1997, 369-380. **[5]** Jacques Patarin, Asymmetric Cryptography with a Hidden Monomial}, CRYPTO 1996: 45-60.

[6] Louis Goubin, Jacques Patarin, Bo-Yin Yang, Multivariate Cryptography. Encyclopedia of Cryptography and Security}, (2nd Ed.) 2011, 824-828.

[7] Sakalauskas., P. Tvarijonas , A. Raulynaitis, Key Agreement Protocol (KAP) Using Con-jugacy and Discrete Logarithm Problema in Group Representation Level}, INFORMATICA, 2007, vol. !8, No 1, 115-124.

[8] D. N. Moldovyan, N. A. Moldovyan, A New Hard Problem over Non-commutative Finite Groups for Cryptographic Protocols, International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security, MMM-ACNS 2010: Computer Network Security, pp. 183-194.

[9] V. Shpilrain, A. Ushakov, The conjugacy search problem in public key cryptography: unnecessary and insufficient,Applicable Algebra in Engineering, Communication and Computing, August 2006, Volume 17, Issue 3–4, pp 285–289.

[10] Delaram Kahrobaei, Bilal Khan, A non-commutative generalization of ElGamal key exchange using polycyclic groups, In IEEE GLOBECOM 2006 - 2006 Global Telecommunications Conference [4150920] DOI: 10.1109/GLOCOM.2006.

[11] Delaram Kahrobaei, Bilal Khan, A non-commutative generalization of ElGamal key exchange using polycyclic groups, In IEEE GLOBECOM 2006 - 2006 Global Telecommunications Conference [4150920] DOI: 10.1109/GLOCOM.2006.

[12] Zhenfu Cao (2012). New Directions of Modern Cryptography. Boca Raton: CRC Press, Taylor & Francis Group. ISBN 978-1-4665-0140-9.

[13] Alexei G. Myasnikov; Vladimir Shpilrain; Alexander Ushakov. Non-commutative Cryptography and Complexity of Group-theoretic Problems. Amer. Math Soc. 2011

[14] J. A. Lopez Ramos, J. Rosenthal, D. Schipani, R. Schnyder, Group key management based on semigroup actions, Journal of Algebra and its applications, vol.16 , 2019.

[15] Gautam Kumar and Hemraj Saini, Novel Noncommutative Cryptography Scheme Using Extra Special Group, Security and Communication Networks ,Volume 2017, Article ID 9036382, 21 pages, https://doi.org/10.1155/2017/9036382.

[16] V. A. Roman'kov, A nonlinear decomposition attack, Groups Complex. Cryptol. 8, No. 2 (2016), 197-207.**27**.

[17] V. Roman'kov, An improved version of the AAG cryptographic protocol, Groups, Complex., Cryptol, 11, No. 1 (2019), 35-42.

[18] A. Ben-Zvi, A. Kalka and B. Tsaban, Cryptanalysis via algebraic span, In: Shacham H. and Boldyreva A. (eds.) Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I, Vol. 10991, 255{274, Springer, Cham (2018).

[19] B. Tsaban, Polynomial-time solutions of computational problems in noncommutative-algebraic cryptography, J. Cryptol. 28, No. 3 (2015), 601-622.

[20] V. Roman’kov, Cryptanalysis of a new version of the MOR scheme, arXiv:1911.00895 [cs.CR].(2019).

[21]V.Roman’kov, Cryptanalysis of two schemes of Baba et al. by linear algebra methods. CoRR abs/1910.09480 (2019).

[22] Max Noether, Luigi Cremona , Mathematische Annalen 59, 1904, p. 1–19.

[[23] V. Ustimenko, On new symbolic key exchange protocols and cryptosystems based on hidden tame homomorphism, Dopovidi. NAS of Ukraine, 2018, n 10, pp. 26-36.

[24] V. Ustimenko, M. Klisowski, On Noncommutative Cryptography with cubical multivariate maps of predictable density, In “Intelligent Computing’’ , Proceedings of the 2019 Computing Conference, Volume 2, Part of Advances in Intelligent Systems and Computing (AISC, volume 99, pp, 654-674.

[25] V. Ustimenko, On desynchronised multivariate algorithms of El Gamal type for stable semigroups of affine Cremona group, Theoretical and Applied Cybersecurity, National Technical University of Ukraine "Igor Sikorsky Kiev Polytechnic Institute", vol 1, 2019, pp. 22-30.

[26] V.Ustimenko, On the usage of postquantum protocols defined in terms of transformation semi-groups and their homomorphisma, Theoretical and Applied Cybersecurity, National Technical University of Ukraine "Igor Sikorsky Kiev Polytechnic Institute", vol 2, 2020, pp. 32-44.

[27] V.Ustimenko, On affine Cremona semigroups, corresponding protocols of Non-commutative Cryptography and encryption with several nonlinear multivariate transformations on secure Eulerian mode. Cryptology ePrint Archive, 2019/1130.

[28] V, Ustimenko, On Multivariate Algorithms of Digital Signatures Based on Maps of Unbounded Degree Acting on Secure El Gamal Type Mode. Cryptology ePrint Archive, 2020/1116.

[29] V. A. Ustimenko, On the extremal graph theory and symbolic computations. Dopov. Nac. Akad. Nauk Ukr., 2013, No. 2, pp. 42-49.

[30] V. Ustimenko, On semigroups of multiplicative Cremona transformations and new solutions of Post Quantum Cryptography, Cryptology ePrint Archive, 133, 2019.

[31] V. Ustimenko, Maximality of affine group, hidden graph cryptosystem and graph’s stream ciphers, Journal of Algebra and Discrete Mathematics, 1, 2004, v.10, pp. 51-65.

**[32]** V. Ustimenko,.On Multivariate Algorithms of Digital Signatures on Secure El Gamal Type Mode. Cryptology ePrint Archive, 2020/984.

[33] V. Ustimenko, On new multivariate cryptosystems based on hidden Eulerian equations, Reports of Nath Acad of Sci, Ukraine, 2017. № 5, pp 17-24.

[34] V..Ustimenko:, On computations with Double Schubert Automaton and stable maps **of** Multivariate Cryptography**.** CoRR abs/2108.08288 (2021).

[35]  I.R. Shafarevich, On some in_nite dimension groups II, Izv. Akad. Nauk SSSR Ser. Mat., Volume 45, No. 1, pp. 214-226 (1981); Mathematics of the USSR-Izvestiya, Volume 18, No. 1, pp. 185-194 (1982).

[36]Yu. Bodnarchuk, Every regular automorphism of the affine Cremona group is inner, Journal of Pure and Applied Algebra, Volume 157, Issue 1, pp. 115-119 (2001).[]31]

[37]V. Ustimenko, On new multivariate cryptosystems based on hidden Eulerian equations, Reports of Nath Acad of Sci, Ukraine, 2017. № 5, pp 17-24.

[[38] V. Ustimenko, On new multivariate cryptosystems based on hidden Eulerian equations over finite fields, Cryptology ePrint Archive, 093, 2017.

[39] V. Ustimenko, Maximality of affine group, hidden graph cryptosystem and graph's stream ciphers, Journal of Algebra and Discrete Mathematics, 2005, v.1, pp 51-65.

[40] V.Ustimenko. On linguistic dynamical systems, graphs of large girth and cryptography, *Journal of Mathematical Sciences*, 140, N3 (2007) 412-434

[41] V. A. Ustimenko.  On extremal graph theory and symbolic computationsОб  экстремально  //Dopovidi NAS of Ukraine, 2012 – №11 – С. 15-21

[42] V. A.,Ustimenko, U. Romanczuk,  On Dynamical Systems of Large Girth or Cycle Indicator and their applications to Multivariate Cryptography, in "Artificial Intelligence, Evolutionary Computing and Metaheuristics ", In the footsteps of Alan Turing  Series: Studies in Computational Intelligence,  Volume 427/2013, 231-256,.

[43] V. Ustimenko, U. Romanczuk, On Extremal Graph Theory, Explicit Algebraic Constructions of Extremal Graphs and Corresponding Turing Encryption Machines, in "Artificial Intelligence, Evolutionary Computing and and Metaheuristics ", In the footsteps of Alan Turing  Series: Studies in Computational Intelligence,  Volume 427/2013, 257-265.