On affine Cremona semigroups, corresponding protocols of Non-commutative Cryptography and encryption with several nonlinear multivariate transformations on secure Eulerian mode.

V. Ustimenko

University of Maria Curie Sklodowska, Lublin 20036, Poland

vasyl@hektor.umcs.lublin.pl

Abstract. We suggest new applications of protocols of Non-commutative cryptography defined in terms of subsemigroups of Affine Cremona Semigroups over finite commutative rings and their homomorphic images to the constructions of possible instruments of Post Quantum Cryptography. This approach allows to define cryptosystems which are not public keys. When extended protocol is finished correspondents have the collision multivariate transformation on affine space K^n or variety $(K^*)^n$ where K is a finite commutative ring and K^* is nontrivial multiplicative subgroup of K.

The security of such protocol rests on the complexity of word problem to decompose element of Affine Cremona Semigroup given in its standard form into composition of given generators. The collision map can serve for the safe delivery of several bijective multivariate maps F_i (generators) on K^n (or $(K^*)^n$) from one correspondent to another. So asymmetric cryptosystem with nonpublic multivariate generators where one side (Alice) knows inverses of F_i but other does not have such a knowledge is possible.

We consider the usage of single protocol or combinations of two protocols with platforms of different nature. The usage of two protocols with the collision spaces K^n and $(K^*)^n$ allows safe delivery of two sets of generators of different nature. In terms of such sets we define an asymmetric encryption scheme with the plainspace $(K^*)^n$, cipherspace K^n and multivariate non-bijective encryption map of unbounded degree $O(n)$ and polynomial density on K^n with injective restriction on $(K^*)^n$. Algebraic cryptanalysis faces the problem to interpolate a natural decryption transformation which is not a map of polynomial density.

Keywords. Multivariate Cryptography, Noncommutative Cryptography, stable transformation groups and semigroups, semigroups of monomial transformations, word problem for nonlinear multivariate maps, hidden tame homomorphisms, key exchange protocols, cryptosystems, linguistic graphs.

1 Introduction.

Investigations of continuous nonlinear transformation of vector spaces R^n and C^n

in term of dynamic systems theory and other method of Chaos Studies have application to Cryptography. The usual scheme use ''discretisation'' of continuous. map, i.e. finding of its natural discrete analog (see [1], [2], [3], [4], [5]). Other approach is connected with studies of K-theory of affine Cremona semigroup of all polynomial maps of affine space K^n into itself, where K is a commutative ring.This is the search for instruments for the constructions of nonlinear maps defined over arbitrary K with special properties. One of the examples is dynamical system of large girth (or large cycle indicator) considered in [6], [7] which allows to introduce large subgroups of cubical transformation on free module K^n. Notice that independently from choice of commutative ring composition of two cubic maps in ''general position'' will have degree 9. So these subgroups are very special sets of transformations. Noteworthy that in the case of commutative ring of characteristic 0 (like fields R and C) there are bijective polynomial maps such that their inverse are not an elements of $S(K^n)$.One of the simplest examples is the map $x \rightarrow x^3$ of one dimensional affine space R. So the family of large subgroups of cubical transformations of K^n, $n>2$ over arbitrary commutative ring is an interesting mathematical object. We believe that studies of corresponding infinite algebraic graphs of large girth defined over commutative rings of characteristic zero is an interesting topic for future investigation, the first results in this direction are presented in [8].

Let symbol $S(K^n)$ stands for the affine Cremona semigroup (see [42]) of all polynomial transformation of K^n. Studies of stable subsemigroups of $S(K^n)$ which are totalities oftransformations of affine space K^n of degree bounded by small constant d are motivated by their cryptographic applications. The cases $d=2, 3$ are of special interest. Notice that $d=1$ corresponds to general affine semigroup $AL_n(K)$ of all transformations of K^n of degree 1. Cryptographic algorithms based on cubical stable semigroups include stream ciphers (see [29] and further references), multivariate Diffie-Hellman key exchange protocols and corresponding El Gamal cryptosystems (see [34] and further references), algorithms of noncommutative cryptography with multivariate platforms ([16], [32],[39],[40]).

Notice that direct usage of cubical transformations from stable semigroups as public encryption instruments does not make sense because the inverse map is also cubical one. One can use $O(n^3)$ pairs of kind plaintext/corresponding ciphertext and interpolate decryption map in time $O(n^{10})$. Anyway for the construction of public keys one can use transformations of stable semigroup in a combination with special unstable transformations (see [9], [10], [30], [31]). For instance in [9] author together with the subgroup of stable cubical subgroup uses other distinguished object which is a totality$^nES(K)$ of nonlinear monomial transformations moving each variable x_i to a single monomial term $t(x_1, x_2,\dots, x_n)$ (algorithms work in the cases $K=F_q$ and $K=Z_m$.In fact subsemigroups of $^nES(K)$ together with stable subsemigroup can be used in secure inverse key exchange protocol in which each correspondents get one element from the pair of polynomial transformations (g, g') from K^m preserving $(K^*)^m$ such that gg' acts on $(K^*)^m$ as identity. Such a protocol developed in a spirit of Noncommutative Cryptography (NC), see ([18]-[28]). In difference with common for NC use of generators and relation we use standard way of Multivariate Cryptography of presenting each element of $S(K^n)$ by its standard form given by lists of monomial terms. Correspondents can use $(K^*)^m$ as plainspace and K^m as cipherspace. So it is an interesting postquantum instrument alternative to public key cryptography stimulated recently by the U.S. NIST step toward mitigating the risk of quantum attacks via the announcement the PQC standardisation process [11]. In March 2019, NIST published a list of candidates qualified to the second round of the PQC process.

We notice that in the cited above studies of usage of stable subsemigroups of $S(K^n)$ for security applications were overlooked. For instance not only inverse but directed tahoma protocols with stable and monomial platforms in tandem can be used for establishment of multivariate asymmetric procedure. We fill this gap in the section 2. Public keys [9], [10] with the usage of semigroup ${}^nES(K)$ and stable subgroups can be used in the case of general commutative ring K(finite or infinite)with nontrivial multiplicative group. This algorithm can be enhanced via algorithms of generation pairs g,g^{-1} from ${}^nES(K)$ with the usage of linguistic graphs defined over commutative group K^*. New version of this cryptosystem is given in section 4. It uses the following scheme. Let as assume that G is a large stable subgroup of $S(K^n)$ with the constant degree d.We generate the composition $z=gf$, where g is a member of mentioned above pair, $f'=TfT'$ where $f \in G$, T and T' are invertible affine transformation from $AL_n(K)$, as public key rules of kind $z_i=z(x_1, x_2,\dots, x_n)$. $z_i \in K[x_1, x_2,\dots, x_n]$. $i=1, 2, ..., n$ in the cases when ground commutative ring K has quite large multiplicative group K^*..In particular we can generate a polynomial transformation z on real vector space R^n, $n>2$ of linear degree and prescribed polynomial density cn^d which preserves $(R^*)^n$ and acts as bijectively on this set (see section 4 of this paper where GJG elements are introduced).Let us assume that commutative ring K is finite and Alice is able to compute f^{1} and g^{-1} in polynomial time.

Public user Bob works with the map of linear degree in variable n which has density $O(n^{d+1})$ (number of monomial terms in all public rules, which coincides with the density of map f' of degree d). This facts guarantee the feasibility of encryption process which consist of computation $c=z(p)$ for element p from the plainspace $(K^*)^n$. Alice in difference with Bob has the factorisation of z into composition of g am f'. She computes $(f')^{-1}(c)=c'$ and restores the plaintext as $g^{-1}(c')$. Notice that unknown for Bob inverse map $(f')^{-1}g^{-1}$ has unbounded degree and exponential density. Thus suggested schemes can be considered in future as candidates for Post Quantum Cryptography (PQC) usage. Notice that this is an algorithm of Multivariate Cryptography with general reference on the complexity to solve nonlinear system of equations. The corresponding system has unbounded degree and corresponding multivariate map is not a bijection. Cryptanalytics can try to factorize this map in a form fg where f is monomial map from ${}^nES(K)$ and g has bounded degree d but general algorithms even subexponential complexity for the completion of this task are unknown.

For proper investigation of these public key algorithms they have to be compared with other known candidates for postquantum usage (like algorithms of the second round of NIST competition).

We discover and alternative option. No need in the announcement of standard form of z publicly because there is a secure way (protocol) for delivery of this multivariate encryption tool for one correspondents to another.In fact instead of z any multivariate map G with injective restriction on $(K^*)^n$ fo linear degree and polynomial density $O(n^d)$, $d=1,2, 3$ can be transported safely from Alice to Bob.

Other option is use a separate delivery of f and g as above which makes the computations faster. Description of the implementations of these delivery algorithms in terms of directed tahoma protocol is given in section 6.

In fact the author of ([14]) noticed that usage of large groups G and ${}^nES(K)$ allows to create natural secure inverse protocol with usage of doubled platform for secure delivery of pairs f^1, g^{-1} (for Alice) and f, g for Bob where f and g written above maps. It means that we can postpone public announcement of gf.

The security of these two solutions with directed and inverse protocols rests on the complexity of decomposition of element of non-commutative subgroup G *of affine Cremona semigroup (*or semigroup ${}^nES(K)$)into the product of several generators given by their standard forms. This is known *word problem* which is unsolvable in polynomial time with usage of Turing machine or Quantum Computer. The first usage of the complexity of word problem for abstract groups was considered in [15].

The further step is presented in section 5 and 6, it brings the option to deliver several bijective multivariate transformations of degree 1, 2 and 3 and conduct algorithm with a governing formal word and hidden multivariate generators.

Stable part of double inverse platforms of [14] constructed in terms of algebraic graphs of geometrical nature, monomial part is defined in terms of parabolic subsemigroup of ${}^nES(K)$ in the cases $K=F_q$ and $K=Z_q$.

In this paper we use double directed tahoma protocol which uses cubical stable groups (section 3) related to constructions of Extremal Group Theory which already were used for the construction of stream ciphers (see [29] and further references) and new subsemigroups of ${}^nES(K)$ (section 4) defined in terms of linguistic graphs over nontrivial multiplicative group K^* of general commutative ring defined in section 3.

2. Some protocols of noncommitative cryptography with multivariate platforms.

Let $S'<S(K^n)$ be a subsemigroup of affine Cremona semigroup and φ be a homomorphism from S' onto semigroup $G<S(K^n)$, $n>m$.

2.1. Additionally we consider a *stable subsemigroupS*, $S'<S<S(K^n)$ and assume thatH is stable semigroup H, $G<H<C(K^m)$. Alice selects elements$s_1, s_2, \ldots ,s_r$, $r>1$ of subsemigroups $S'$and computes $\varphi(s_i)=u_i$. She takes invertible elements $h\in S(K^n)$ of kind av, $deg(a)=1$, $v\in S$and $f\in C(K^n)$, $f=bg$,$deg(b)=1$, $g\in H$ and forms pairs ($a_i=hs_ih^{-1}$, $b_i=f u_if^1$) and sends them to Bob.

He forms word $w=(a_{i(1)})^{\alpha(1)}(a_{i(2)})^{\alpha(2)}\ldots (a_{i(t)})^{\alpha(t)}$, $t>r-1$, $i(j)\in\{1,2,\ldots,r\},\alpha(j)>0,j=1,2,\ldots,t$ and sends it to Alice. Bob changes alphabet via the substitution of b_i instead ofa_i and keeps the word $u=(b_{i(1)})^{\alpha(1)}(b_{i(2)})^{\alpha(2)}\ldots (b_{i(t)})^{\alpha(t)}$.

Alice computes u as $f\varphi(h^{-1}wh)f^{-1}$.

So Alice and Bob when the protocol ends have collision transformation of the affine space K^m.

Examples of the implemetations of this algorithm can be found in [16].

2.2. Let us consider above algorithms in the case when semigroup$S$ consists on *toric elements* and $H<^mEG(K)$ and $S=S'$.

Alice forms h and h^{-1} from $^nEG(K)$ together with pair f,f^{-1} from $^mEG(K)$ and proceed with the modification of previous algorithm.

Alice selects elementss_1, s_2, … ,s_r , $r>1$ of semigroups S and computes $\varphi(s_i)^{-1} = u_i$. She takes invertible elements h and f to form pairs ($a_i=hs_ih^{-1}$, $b_i=f u_if^{1}$) and sends them to Bob. The rest of the algorithm is identical to case of procedure 2.1.

After the completion of this protocol Alice and Bob have common maps $u$acting on the variety$(K^*)^m$.

Security base: The adversary has to solve the ***word problem*** for the subsemigroupS', i. e., find the decomposition of w from S'into generators a_i, $i= 1, 2,...,t$. The general algorithm to solve this problem in polynomial time for the variable $n$is unknown, as well as a procedure to get its solution in terms of quantum computations. The problem depends heavily on the choice of group.

Remark. Of course in each case alternative ways of computation of the value $\sigma(w)$ of isomorphism σ between semigroup $<a_1, a_2, ...,a_r>$ and group $<b_1, b_2, ...,b_r>$ given by the rule$\sigma(a_i)= b_i$ have to be investigated.

2.3. On platforms acting in tandem.

2.3.1. Alice and Bob use algorithm 2.1 with the output u on K^m as leading procedure. Supporting procedure is algorithm of kind 2.2 with the same commutative ring K and parameter m. Alice uses platform of algorithm 2.1 and generates elements v and v^{-1}.She keepsv^{-1} fo herself and send v+u to Bob. So Bob gets v. Alice selects the input of 2.2 for her correspondent as a_i,b_i , $i=1,2,..., r'$. She sends pairs $(a_i, , v^{-1}(b_i))$. Notice that the elements $v^{-1}(b_i)$) are well defined maps of K^m into K^m, they have polynomial density.

Bob computes pairs (a_i,b_i) because of his/her possession of v. After the completion of supporting procedure Alice and Bob get common elements $z$of $^mEG(K)$.

Additionally Alice generates elements y and y^{-1} of $^mEG(K)$. She keeps y^{-1} for herself. She takes z of kind $x_i \rightarrow z_i(x_1, x_2,\dots, x_m)$, $i=1,2,...,m$ and forms the tuple $(z_1y_1, z_2y_2,...,z_my_m)$ to send it to Bob. Coordinates of the tuple are computed via multiplication of monomial expressions in $K[x_1, x_2,\dots, x_m]$. Thus Bob computes map y easily.

They use $(K^*)^m$ as plainspace and K^m as cipherspace.

To encrypt Alice maps her message p in the alphabet K^* to $y^{-1}(p)=m$ and then she computes the ciphertext $c= v^{-1}(m)$.

Bob decrypts via application of v *to* c and computation of $y(v(c))$.

Similarly Bob encrypts p via consecutive computation of y and $v(y(p))$.

Alice applies v^{-1} to ciphertext c and computes the plaintext as$y^{-1}(v^{-1}(c))$.

Remark. Encryption and decryption functions of the above algorithm can be treated as polynomial maps of K^m to K^m because elements of ${}^{m}EG(K)$ act naturally on K^m. Between encryption and decryption functions there is a density gap because decryption map is not a transformation of polynomial density. Such pairs can be used as non-bijective stream ciphers in a spirit of [29]. In the tandem procedure interception of plaintexts with corresponding ciphertext attacks are unfeasible without the computation of $\sigma(w)$.

2.3.2 Alice and Bob can use algorithm 2.2 with collision map u on $(K^*)^m$ as leading procedure. Supporting procedure is algorithm of kind2.1 with the same commutative ring K and parameter m. Alice creates elements z and z^{-1} *of* ${}^{m}EG(K)$. She takes z of kind $x_i \rightarrow z_i\ (x_1, x_2,\dots, x_m)$, $i=1,2,\dots,m$ and forms the tuple $(z_1u_1, z_2u_2,\dots,z_mu_m)$ to send it to Bob. He uses his knowledge on u to compute z.

Alice sets pairs (a_i, b_i) to start supporting protocol 2.1. She sends $b_i(z^{-1})$ which has polynomial density to Bob. He uses his knowledge on z and computes b_i. Correspondents execute protocol 2.1 and get collision stable map u.

Alice uses platform of 2.1 to generate mutually invertible transformations y and y^{-1}acting on K^m. She keeps y^{-1} for herself and sends $y+u$ to Bob. He subtracts u and gets y.

As in previous algorithm Alice and Bob use plainspace $(K^*)^m$ and ciphertext K^m.

To encrypt Alice maps her message p in the alphabet K^* to $z^{-1}(p)=m$ and then she computes the ciphertext $c= y^{-1}(m)$.

Bob decrypts via application of y *to* c and computation $z^{-1}(y(c))$.

Similarly Bob encrypts p *via* consecutive computation of z *to* p and $y(z(p))$.

Alice applies y^{-1} to ciphertext c and computes the plaintext as $z^{-1}(y^{-1}(c))$.

Remark. In the case 2.2 Alice (or Bob) instead of mutually invertible y, y^{-1} can use elements $w, w' \in S((K)^m$ of polynomial density such that theiry^{-1}restrictions on $(K^*)^m$ are injective maps to K^m and composition ww' acts on $(K^*)^m$ as identical map. Algorithm of generation such pairs is introduced in [14], [29], [30] and [31].

Algorithms of generation of pairs (z, z^{-1}) from ${}^{m}EG(K)$ are described in [32].

3. On linguistic and extremal graphs and stable nonlinear subgroups of affine Cremona group.

3.1. Some definitions of Extremal graph theory.

The missing definitions of graph-theoretical concepts in the case of simple graphs which appears in this paper can be found in [33]. All graphs we consider are simple ones, i. e. undirected without loops and multiple edges.

When it is convenient, we shall identify Γ with the corresponding antireflexive binary relation on $V(\Gamma)$, i.e. $E(\Gamma)$ is a subset of $V(\Gamma) \times V(\Gamma)$. The girth of a graph Γ, denoted by $g = g(\Gamma)$, is the length of the shortest cycle in Γ. The diameter $d= d(\Gamma)$ of the graph Γ is the maximal length of the shortest pass between its two vertices. Let $g_x= g_x(\Gamma)$ be the length of the minimal cycle through the vertex x from the set $V(\Gamma)$

of vertices in graph Γ.We refer to $Cind(\Gamma)=max(g_x|$ x from $V(\Gamma))$ as cycle indicator of the graph.

The family Γ_i of connected k-regular graphs of constant degree is a family of small world graphs if $d(\Gamma_i)\leq clog_k(v_i)$, for some constant c, $c>0$.

Recall that family of regular graphs $\Gamma_i$of degree$k$ and increasing order $v_i$is a family of graphs of large girth if $g(\Gamma_i)\geq clog_k(v_i)$, for some independent constant c, $c>0$.

We refer to the family of regular simple graphs $\Gamma_i$of degree $k$and order $v_i$as family of graphs of large cycle indicator, if $Cind(\Gamma_i)\geq clog_k(v_i)$ for some independent constant c, c >0.

Notice that for vertex-transitive graph its girth and cycle indicator coincide. Defined above families plays an important role in Extremal Graph Theory, Theory of LDPC codes and Cryptography.(see [34] and further references).

3.2. The algebraic graphs *A(n, K)* and *D(n,K),* some results and open questions.

Below we consider the family of graphs $A(n,K)$ and $D(n,K)$, respectively where $n>5$ is a positive integer and K is a commutative ring. In the case of $K=F_q$ we use symbols $A(n, q)$ and $D(n,q)$ for these graphs.efine these graphs as homomorphic images of infinite bipartite graphs $A(K)$ and $D(K)$ for which partition sets P and L formed by 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 ill be called points and those of L lines. To distinguish points from lines we use parentheses and brackets. The description is based on the connections of these graphs with Kac-Moody Lie algebra with extended diagram A_1. The vertices of $D(K)$ are infinite dimensional tuples over K. We write them in the following way $(p)=(p_{0,1}, p_{1,1}, p_{1,2}, p_{21}, p_{22}, p'_{22}, p_{23}, \ldots, p_{i,i}, p'_{i,i}, p_{i,i+1}, p_{i+1,i}, \ldots)$, $[l]=[l_{1,0}, l_{1,1}, l_{1,2,}, l_{21}, l_{22}, l'_{22}, l_{23}, \ldots, l_{i,i}, l'_{i,i}, l_{i,i+1}, l_{i+1,i}\ldots]$. We assume that almost all components of points and lines are zeros. The condition of incidence of point (p) and line $[l]$ ($(p)I[l]$) can be written via the list of equations below.

$l_{i,i}-p_{i,i}=l_{1,0}p_{i-1,i}$; $l'_{i,i}-p'_{i,i}=l_{i,i-1}p_{0,1}$; $l_{i,i+1}-p_{i,i+1}=l_{i,i}p_{0,1}$;$l_{i+1,i}-p_{i+1,i}=l_{1,0}p'_{i,i}$ .This four relations are defined for $i\geq 1$, ($p'_{1,1}=p_{1,1}$, $l'_{1,1}=l_{1,1}$) .

Similarly we define graphs $A(K)$ on the vertex set consisting of points and lines
$(p)=(p_{0,1}, p_{1,1}, p_{1,2}, p_{21}, p_{22,}, p_{23}, \ldots, p_{i,i}, p_{i,i+1,\ldots})$,
$[l]=[l_{1,0}, l_{1,1}, l_{1,2,}, l_{21}, l_{22}, l_{23}, \ldots, l_{i,i}, l_{i,i+1,\ldots}]$ such that point (p) is incident with the line $[l]$ $((p)I[l]$, if the following relations between their coordinates hold: $l_{i,i}-p_{i,i}=l_{1,0}p_{i-1,i}$; $l_{i,i+1}-p_{i,i+1}=l_{i,i}p_{0,1}$.

We consider graphs $A(n,K^*)$ and $D(n, K^*)$ with partition sets isomorphic to $(K^*)^n$ given by equations of $A(n,K)$ and $D(n,K)$ where operation'' – ''is changed for division /. It is clear that the set of indices $A=\{(1; 0), (0; 1), (1; 1), (1; 2), (2; 2), (2; 3), \ldots,$ (i-1, i), (i, i),… } is a subset in $D=\{(1, 0), (0; 1), (1, 1), (1, 2), (2; 2), (2, 2)', \ldots, (i-1, i); (i; i-1); (i, i); (i, i)', \ldots\}$. Points and lines of $D(K)$ (or $D(K^*)$ are functions from $K^{D-\{(1,0)\}}$and $K^{D-\{(0,1)}$ (or $(K^*)^{D-\{(1,0)\}}$ and $(K^*)^{D-\{(0,1)}$) and their restrictions on $A-\{(1,0)\}$ and $A-\{(0,1)\}$ define homomorphism Ψ of graph $D(K)$ onto $A(K)$ (or $D(K^*)$ and $A(K^*)$).

For each positive integer $m\geq 2$ we consider subsets $A(m)$ and $D(m)$ containing first $m+1$ elements of A and D with respect to the above orders .

Restrictions of points and lines of $D(K)$ (or $D(K^*)$)onto $D(m)-\{(1,0)\}$ and $D(m)-\{(0,1)\}$ define graph homomorphism ${}^D\Delta(m)$ with image denoted as $D(n, K)(D(n,$

K)).* Similarly restrictions of points and lines of *A(K) (or A(K*)* onto *A(m) -{(1,0)}* and *A(m)-{(0,1)}* defines homomorphism ${}^{A}\Delta(m)$ of graph *A(K) (or A(K*))* onto graph denoted as *A(m, K) (A(m ,K*) respectively).*

We also consider the map $\Delta(m)$ on vertices of graph *D(m, K)* (or *D(m,K*)*) sending its point $(p) \in K^{D(m)-\{(1,0)\}}$ (or $(K^*)^{D(m)-\{(1,0)\}}$) to its restriction into $D(m) \cap A-\{(1,0)\}$ and its line $[l] \in K^{D(m)-\{(0,1)\}}$ (or $(K^*)^{D(m)-\{(0,1)\}}$) to its restriction onto $D(m) \cap A-\{(0,1)\}$. This map is homomorphism of *D(m, K)* onto *A(n, k)*, $n=|D(m) \cap A|-1$ or *D(m,K*)* onto *A(n,K*).*

Graph $D(q)=D(F_q)$ is *q*-regular forest. Its quotients *D(n, q)* are edge-transitive graphs. So their connected components are isomorphic. Symbol *CD(n, q)* stands for the graph which is isomorphic to one of such connected components.

Family CD(n, q), n=2,3,...is a family of large girth for each fixed parameter q, q>2 and n=2,3, ... (see [35] and further references).

The question '*'Whether or not CD(n, q) is a family of small world graphs''* is still open''.

Graph A(q), q>2 is a q-regular tree. Graphs A(n, q) are not vertex transitive. They form a family of graphs with large cycle indicator, which is q-regular family of small world graphs [36].

The question '*'Whether or not A(n,q), n=2,3,... is a family of large girth''* is still open.

We hope that introduced above graphs $A(n, F_q{}^*)$ and $D(n, F_q{}^*)$ possess interesting extremal and spectral properties

Groups *GD(n, K)* and *GA(n, K)* of cubical transformations of affine space K^n associated with graphs *D(n, K)* and *A(n, K)* are interesting objects of algebraic transformation group theory because of composition of two maps of degree 3 for vast majority of pairs will have degree 9. Constructions and applications of these families of transformations groups are recently observed in [37] where some extensions of these groups are introduced.

3.2. Transformation groups related to algebraic graphs *A(n, K)* and *D(n, K)*.

All graphs defined in section 2 belong to class *L* of linguistic graphs $\Gamma=\Gamma(K)$ of type *(1, 1, n-1)*, $n \in N$ or $n=\infty$.defined over commutative ring *K* which contains bipartite graphs with the point set $P=K^n$ and line set $L=K^n$ such that $(p)=(p_1, p_2, \dots ,p_n) \in P_n$ and $[l]=[l_1, l_2, \dots , l_n] \in L_n$ form an edge of Γ if the following conditions holds ${}^{2}ap_2 - {}^{2}bl_2={}^{2}f(l_1, p_1)$, ${}^{3}ap_2-{}^{3}bl_2={}^{3}f(p_1, p_{12}, l_1 , l_2)$, ... , ${}^{n}ap_n-{}^{n}bl_n={}^{n}f(p_1, p_2, \dots, p_n, l_1, l_2, \dots , l_n)$, where ${}^{i}a$ and ${}^{i}b$, $i \geq 2$ are elements of multiplicative group K^* and f_i are multivariate polynomials. (see [38]. [6]). We define *colours* $(\rho((p))$ and $\rho([l])$ of the point *(p)* and the line *[l]* as their first coordinates p_1 and l_1. We introduce well defined operator *N(v, a)* of computing the neighbour of vertex *v* of colour $a \in K$ and *colour jump* operator *J(v,a)* sending point or line $v=(v_1, v_2, \dots, v_n)$ to $u=(a, v_2, v_3, \dots, v_n)$.

Let $S(K^n)$ stands for the Cremona semigroup of polynomial transformations of free module K^n and $C(K^n)$ be affine Cremona group of invertible elements of $S(K^n)$ with the polynomial inverse. These algebraic structures are important objects of algebraic geometry. One of the difficult problem is about constructions of families of stable subgroups G_n of $C(K^n)$ (or semigroup S_n of $S(K^n)$) i. e groups of polynomial transformation with maximal degree equals to constant c. Notice that for the majority of pairs

f, $g \in C(K^n)$ of degrees r and s their composition has degree rs. So this problem is difficult, it has strong cryptographical motivations.

We consider totality $St(K)$ of strings of kind $(f_1, f_2, \dots, f_k)$ where $f_i \in K[x]$. We will identify polynomial f and the map $x \to f(x)$ from $S(K)$. The product of two chains $(f_1, f_2, \dots, f_k)$ and $(g_1, g_2, \dots, g_t)$ is the chain $(f_1, f_2, \dots, f_k, g_1(f_k), g_2(f_k), \dots, g_t(f_k))$. Empty string is the unity of semigroup $St(K)$. In fact $St(K)$is a semidirect product of a free semigroup over the alphabet $K[x]$ and Cremona semigroup $S(K)$. We refer to $St(K)$ as semigroup of polynomial strings. Let $St'(K)$ stands for the semigroup of strings of even length from $St(K)$ and $\sum(K)$ be subsemigroups of strings of even length with coordinates of kind $x+c$, $c \in K$. Let $u=(f_1, f_2, \dots, f_k)$ be an element of $St'(K)$ and $x \to f_k(x)$ is an element of $C(K)$. We refer to $rev(u)= (f_{k-1}(f_k^{-1}(x)). f_{k-1}(f_k^{-1}(x)) \dots, f_1(f_k^{-1}(x)).f_k^{-1}(x))$ as reverse string to u.

In the case of linguistic graph $\Gamma=\Gamma(K)$ *of type (1,1,n-1)* the path consisting of its vertices $v_0, v_1, v_2, \dots, v_k$ is uniquely defined by initial vertex v_0, and colours $\rho(v_i)$, $i=1, 2, \dots, k$ of other vertices from the path. We can consider graph $\Gamma'=\Gamma(K[x_1, x_2, \dots, x_n])$ defined by the same with Γ equations but over the commutative ring $K[x_1, x_2, \dots, x_n]$.

So the following symbolic computation can be defined. Take the *symbolic point* $x=(x_1, x_2, \dots, x_n)$, where x_i are generic variables of $K[x_1, x_2, \dots, x_n]$ and polynomial string $C \in St'(K)$ which is a tuple of polynomials $f_1, f_2, \dots, f_k$, from $K[x_1]$ with even parameter k. $(x=x_1)$. Form the path of vertices $v_0, =x$, v_1 such that $v_1 I v_o$ and $\rho(v_1)=f_1(x_1)$, v_2 such that $v_2 I v_1$ and $\rho(v_2)=f_2(x_1)$, ..., v_k such that $v_k I v_{k-1}$ and $\rho(v_k)=f_k(x_1)$. We choose parameter k as even number. So v_k is the point from the partition set $K[x_1, x_2, \dots, x_n]^n$ of the graph Γ'.

We notice that the computation of each coordinate of v_i depending on variables $x_1, x_2, \dots, x_n$ and polynomials $f_1, f_2, \dots, f_k$ needs only arithmetical operations of addition and multiplication. As it follows from the definition of linguistic graph final vertex v_k (point) has coordinates $(h_1(x_1), h_2(x_1,x_2), h_3(x_1,x_2,x_3), \dots, h_n(x_1,x_2, \dots, x_n))$, where $h_1(x_1)=f_k(x_1)$. Let us consider the map ${}^{\Gamma}H(C)$: $x_i \to h_i(x_1, x_2, \dots, x_n)$, $i=1, 2, \dots, n$ which corresponds to polynomial string C.

Proposition 1. *The map* ${}^{\Gamma}\eta : C \to {}^{\Gamma}H(C)$ *is a homomorphism of* $St'(K)$ *into Cremona semigroup* $S(K^n)$.

Lemma 1 . *Let* $u=(f_1, f_2, \dots, f_k)$ *and* $x \to f_k(x)$ *is an element of* $C(K)$.*Then for each linguistic graph* Γ *of type (1, 1,n-1) element* $rev(u)u$ *be an element of kernel of* ${}^{\Gamma}\eta$.

More general form of this statement is proven in [14].

We refer to ${}^{\Gamma}\eta$ as linguistic compression map. If K is finite then the map converts totality of potentially infinite strings into finite semigroup.

Theorem 1.*If* Γ *is one of graphs* $D(n, K)$ *and* $A(n,K)$ *, then* ${}^{\Gamma}\eta(\sum(K))$ *is stable subgroup of*$C(K^n)$ *of degree 3.*

We denote ${}^{\Gamma}\eta(\sum(K))$ for $\Gamma=D(n,K)$ and $\Gamma=A(n, K)$ as $GD(n\ K)$ and $GA(n, K)$.These groups were already used in all cryptographical applications of graphs $D(n, K)$ and $A(n,K)$.

Proposition 2. *Homomorphisms* δ *of* $D(n,K)$ *onto* $A(m,K)$, $n>m$ *described in section 2 induces homomorphism* $ind(\delta)$ *of* $GD(n,K)$ *onto* $GA(m,K)$, $n>m$.

4. On Eulerian groups and semigroups and multiplicative linguistic graphs.

4.1 Basic constructions.

Similarly to the case of commutative ring we introduce a linguistic graph $I= \Gamma(G)$ over abelian group G defined as bipartite graph with partition sets isomorphic to G^n such that $(x_1, x_2, ..., x_n)\ I\ [y_1, ,y_2, ..., y_n]$ if and only if $x_2/y_2=g_2w_2(x_1, ,y_1)$, $x_3/y_3=g_3w_3(x_1, x_2, y_1, y_2)$, …, $x_n/y_n=g_nw_n(x_1, x_2, ..., x_{n-1}, y_1, y_2, ..., y_{n-1})$, where $g_i \in G$, $i\geq 2$ and w_i are words in characters x_i and y_j from G. We define *colours* $\rho((p))$ and $\rho([l])$ of the point (p) and the line $[l]$ as their first coordinates p_1 and l_1. We introduce well defined operator $N(v, a)$ of computing the neighbour of vertex v of colour $a\in G$.

Let K be a finite commutative ring with the multiplicative group K^* of regular elements of the ring. We take Cartesian power ${}^nE(K) =(K^*)^n$ and consider an Eulerian semigroup ${}^nES(K)$ of transformations of kind $x_1 \rightarrow d_1x_1^{a(1,1)}x_2^{a(1,2)} \ldots x_m^{a(1,n)}$, $x_2 \rightarrow d_2x_1^{a(2,1)}x_2^{a(2,2)} \ldots x_m^{a(2,n)}$,…,$x_m \rightarrow d_nx_1^{a(n,1)}x_2^{a(n,2)} \ldots x_m^{a(n,n)}$,where $a(i,j)$ are elements of arithmetic ring Z_d, $d=|K^*|$, $d_i\in K^*$.

Let ${}^nEG(K)$ stand for Eulerian group of invertible transformations from ${}^nES(K)$. 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.

The problems of constructions of large subgroups G of ${}^nEG(K)$, pairs (g, g^{-1}), $g \in G$, and tame Eulerian homomorphisms$E$: $G\rightarrow H$, i. e. computable in polynomial time $t(n)$ homomorphisms of subgroup G of ${}^nEG(K)$ onto $H< {}^mEG(K)$ are motivated by tasks of Nonlinear Cryptography .

We consider totality $St(K^*)$ of strings of kind $(f_1, f_{2,\ldots,} f_{k,})$ where f_i are expressions of kind ax^d,$d\in Z_m$, $m=|K^*|$, $a\in K^*$. We will identify polynomial f and the map $x\rightarrow f(x)$ on K^*. The product of two chains $(f_1, f_{2,\ldots,} f_{k,})$ and $(g_1, g_{2,\ldots,} , g_{t,})$ is the chain $(f_1, f_{2,\ldots,} f_{k,}$, $g_1(f_k\ ),g_{2,}(f_k\ ),\ _{\ldots,}\ g_{t,}(f_k\ ))$. Empty string is the unity of semigroup $St(K^*)$. Let $St'(K^*)$ stand for the semigroup of strings of even length from $St(K^*)$ and $RS(K^*)$ stand for totality of strings$(f_1, f_{2,\ldots,}f_{k,})$ with invertible maps $x\rightarrow f_{k,}(x)$ from $St'(K^*)$.We refer to elements of $RS(K^*)$ as reversible multiplicative strings.

Let $K^*[x_1, x_2, ..., x_n]$ be group of monomials from $K[x_1, x_2, ..., x_n]$. with operation of multiplication. For each linguistic graph $\Gamma(K^*)$ over K^* we can consider infinite graph $\Gamma'=\Gamma(K^*[x_1, x_2, ..., x_n])$ defined by the same equations with Γ but over the commutative group $K^*[x_1, x_2, ..., x_n]$.

So the following symbolic computation can be defined. Take the *symbolic point* $x=(x_1, x_2, ..., x_n)$, where x_i are generic variables of $K^*[x_1, x_2, ..., x_n]$ and polynomial string $C\in St'(K^*)$ which is a tuple of polynomials $f_1,, f_2,, \ldots , f_k$, from $K^*[x_1]$ with even parameter k $(x=x_1)$. Form the path of vertices $v_{0,}=x$, v_1 such that v_1Iv_o and $\rho(v_1)=f_1(x_1)$, v_2 such that v_2Iv_1 and $\rho(v_2)=f_2(x_1)$, ..., v_k such that v_kIv_{k-1} and $\rho(v_k)=f_k(x_1)$. We choose parameter k as even number. So v_k is the point from the partition set $K^*[x_1, x_2,\ldots, x_n]^n$ of the graph Γ'.

As it follows from the definition of linguistic graph final vertex v_k (point) has coordinates $(h_1(x_1), h_2(x_{1,}x_2), h_3(x_{1,}x_2,x_3),\ldots,h_n(x_{1,}x_2,\ldots, x_n))$, where $h_1(x_1)=f_k(x_1)$. Let us consider the map ${}^\Gamma H^*(C)$: $x_i\rightarrow h_i(x_1, x_{2,}\ldots, x_n)$, $i=1, 2,\ldots, n$ which corresponds to polynomial string C.

Proposition 3. *For each linguistic graph Γ over K^* the map ${}^\Gamma\eta^* :C \rightarrow {}^\Gamma H^*(C)$ is a homomorphism of $St'(K^*)$ into Eulerian semigroup ${}^nES(K)$.*

We refer to ${}^\Gamma\eta$ * as linguistic multiplicative compression map.

Proposition4. *For each linguistic graph Γ over K^* the image* ${}^{\Gamma}\eta^*(RS(K^*))$ is *a subgroup of Eulerian group* ${}^{n}EG(K)$.

We denote ${}^{\Gamma}\eta(RS(K^*))$ for $\Gamma=D(n,K^*)$ and $\Gamma=A(n,K^*)$ as $GD(n,K^*)$ and $GA(n,K^*)$.

Proposition 5. *Homomorphisms δ of $D(n,K^*)$ onto $A(m,K^*)$, $n>m$ described in section 2 induces tame Eulerian homomorphism of group $GD(n,K^*)$ onto $GA(m,K^*)$, $n>m$.*

Let π and δ be two permutations on the set $\{1,2,\dots, n\}$. Let us consider a transformation of $(K^*)^n$, $K=Z_m$or $K= F_q$and $d=|K^*|$. We define transformation ${}^{A}JG(\pi, \delta)$, 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.

$$y_{\pi(1)}=\mu_1 x_{\delta(1)}^{a(1,1)}$$
$$y_{\pi(2)}=\mu_2 x_{\delta(1)}^{a(2,1)} x_{\delta(2)}^{a(2,2)}$$
$$\dots$$
$$y_{\pi(n)}=\mu_n x_{\delta(1)}^{a(n,1)} x_{\delta(2)}^{a(n,2)} \dots x_{\delta(n)}^{a(n,n)}$$

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

We refer to ${}^{A}JG(\pi, \delta)$ as Jordan Gauss multiplicative transformation or simply JG element. It is an invertible element of ${}^{n}ES(K)$ with the inverse of kind ${}^{B}JG(\delta, \pi)$ such that $a(i,i)b(i,i)=1 \ (mod\ d)$. Notice that in the case $K= Z_m$straightforward process of computation of the inverse of JG element is connected with 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.

We introduced Generalized Jordan Gauss elements (GJG-transformations)of $S(K^n)$ in the case of arbitrary commutative ring with nontrivial multiplicative group.

For this task we consider the totality $I(K)$ of Eulerian positive integers e such thatequation $x^e=b$ where $x \in K^*$, $b \in K^*$ has a unique solution and change condition $(a(1,1),d)=1$, $(a(2,2),d)=1,\dots,(a(n,n),d)=1$ in the definition of JG element for $a(i,i)\in I(K)$.

Noteworthy that such generalization is especially productive in the case of infinite rings. We refer to the composition of several GJG elements as computationally tame multiplicative transformation.

Let ${}^{n}ES'(K)$ stands for the group of computationally tame elements from ${}^{n}ES(K)$.

4.2. Generalizations.

We consider totality $BS(K^*)$ of strings of kind $(f_0, f_1, f_{2,\dots,} f_{k,})$ where f_i are expressions of kind ax^d, $d\in Z_m$, $m=|K^*|$, $a\in K^*$ and $k=0(mod\ 4)$. We will identify polynomial f and the map $x\rightarrow f(x)$ on K^*. The product of two chains $(f_0 f_1, f_{2,\dots,} f_{k,})$ and $(g_0, g_1, g_{2,\dots,}, g_{t,})$ is the chain $(f_0 f_1, f_{2,\dots,} f_{k-1,}, g_0(f_k), g_{1,}(f_k), \dots, g_{t-1,}(f_k), g_t(f_k))$. The string of kind (e), where e is identity map $x\rightarrow x$ is the unity of semigroup $BS(K^*)$. Let $BR(K^*)$ stand for totality of strings$(f_1, f_{2,\dots,} f_{k,})$ from $BS(K^*)$ with invertible maps $x\rightarrow f_{k,}(x)$ from $EG(K^*)$.We refer to elements of $BR(K^*)$ as reversible multiplicative strings. Let $u=(f_1, f_{2,\dots,} f_{k,})$ be an element of $BR(K^*)$. We refer to string $rev(u)=(f_{k-1,}(f_{k,}^{-1}), f_{k-1,,}(f_{k,}^{-1}),,\dots, f_1(f_{k,}^{-1}), f_{k,}^{-1})$ as reverse string for u. Let $K^*[x_1, x_2, \dots, x_n]$ be group of monomials from $K[x_1, x_2, \dots, x_n]$ with operation of multiplication. For each linguistic graph $\Gamma(K^*)$ over K^* of type $(1, 1, n-$

1) we can consider infinite graph $\Gamma'=\Gamma(K^*[x_1, x_2, ,..., x_n])$ defined by the same equations with Γ but over the commutative group $K^*[x_1, x_2, ..., x_n]$.

Let us consider the homomorphism of the group $BS(K^*)$ into Cremona semigroup $S(K^n)$ defined in terms of linguistic graph $I=I^n(K^*)$. Notice that one can consider graph $I^n(K')$ over the extension K' of K^* with the usage of the same equations. Let us take $K'=K^*[x_1, x_2,..., x_n]$,where x_i are formal variables and consider an infinite graph $I^n(K^*[x_1,x_2,...,x_n])$, with partition sets $P'=K^*[x_1,x_2,..., x_n]^n$ and $L'=K^*[x_1, x_2,..., x_n]^n$. After that we take a bipartite string $u=(f_0, f_1, f_2, f_3, f_4, f_5,f_6,..., f_{t-1}, f_t)$ formed by a totality of terms from the subgroup $K^*[x_1]$ of $K'=K^*[x_1, x_2,...,x_n]$ and the point $(x)=(x_1, x_2,..., x_n)$ formed by generic elements of K'. This data defines uniquely a *skating chain(x),* $J((x),f_0)=({}^1x),N(({}^1x),f_1)=[{}^2x],J([{}^2x],f_2)=[{}^3x],N([{}^3x],f_3)=({}^4x),J(({}^4x)$, $f_4)=({}^5x),..., J([{}^{t-2}x], f_{t-2})=[{}^{t-1}x],N([{}^{t-1}x],f_{t-1})=({}^tx),J(({}^tx),f_t)=({}^tx)$.

Let $({}^tx)$be the tuple $(f_t, F_2, F_3,...,F_n)$ where $F_i \in K^*[x_1, x_2,..., x_n]$. We define ${}^I\Psi(u)$ as the map $(x_1, x_2,..., x_n)\rightarrow(H_t, F_2, F_3,...,F_n)$ and refer to it *as chain transition of point variety.*

The statement written below follows from the definition of the map.

Lemma 2. *Let $I(K^*)$be a linguistic graph of type (1, 1, n-1) over K^* defined over multiplicative group of commutative ring K.* **Then** *map* $\boldsymbol{\psi={}^I\psi}$**:**$BS(K^*)\rightarrow {}^nES(K^*)$ *is a homomorphism of semigroups.*

Lemma 3. Let *$u\in BR(K^*)$ then u rev(u) is an element of kernel of* ${}^I\boldsymbol{\psi}$**,**

Corrolary. ${}^I\boldsymbol{\psi}(BR(K^*))$ *is a subgroup of* ${}^I\boldsymbol{\psi}(BS(K^*))$

Generalisation of lemma 1 for the case of general linguistic graph over commutative group is proposed in [14].

Let $ED(n, K^*)$ and $EA(n, K^*)$ stands for ${}^I\boldsymbol{\psi}(BS(K^*))$ with $I=D(n, K^*)$ and $I=A(n,K^*)$. It is easy to see that$ED(n, K^*)>GD(n, K^*)$ and $EA(n, K^*)>GA(n, K^*)$.

Below we define an extension of group of computationally tame transformations.

4.3. On general linguistic graphs over commutative groups and generating procedure of mutually inverse transformations of $(K^*)^n$.

Similarly to the case of commutative ring we introduce a linguistic graph $I(G)=\Gamma(G)$ over abelian group G defined as bipartite graph with partition sets $P=P_{s,m}=G^{s+m}$ and $L=L_{r,m}=G^{r+m}$ such that $x=(x_1, x_2,...,x_s, x_{s+1}, x_{s+2}, ..., x_{s+m})Iy=[y_1, y_2, ... , y_r, ,y_{r+1},y_{r+2} , ..., y_{r+s}]$ if and only if $x_2/y_2=g_2w_2(x_1, ,y_1)$, $x_3/ y_3=g_3w_3(x_1, x_2, y_1, y_2)$, ..., $x_n/y_n=g_nw_n(x_1, x_2, ..., x_{n-1}, y_1, y_2, ..., y_{n-1})$, where $g_i \in G$, $i\geq 2$ and w_i are words in characters x_i and y_j from G. We refer to the triple (r, s, m) as type of $I(G)$.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 $\boldsymbol{a}=(p_1, p_2, ..., p_s)$ or $\boldsymbol{a}=(l_1, l_2 , ..., l_r)$ and introduce well defined operator $N(v, a)$ of computing the neighbour of vertex v of colour $a\in G^s$ or $a\in G^r$. Similarly to the case of linguistic graph over commutative *ring* we define jump operator $J(p, a)$, $a\in K^s$ on partition set P and $J(l,a)$, $a\in K^r$ on partion set L by conditions $J(p,a)=(a_1, a_2, ... a_s, p_{1+s}, p_{2+s}, ..., p_{s+n})$and $\rho(J(l,a))=[a_1, a_2, ...a_r, p_{1+r}, p_{2+r}, ..., p_{r+m}]$.

Let as assume that $G=K^*$ and consider semigroup ${}^sS^r(K^*)$ of tuples $F=(f_1(x_1, x_2,... ,x_s), f_2(x_1, x_2,... , x_s),..., f_r(x_1, x_2,... , x_s))$ where $f_i(x_1, x_2,... ,x_s)$ are monomial terms with coefficients from K^*.We identify elements F of ${}^sS^s(K^*)$ with the maps $\alpha(F):x_1\rightarrow f_1(x_1, x_2,... ,x_s)$, $x_2\rightarrow f_2(x_1, x_2,... ,x_s),...,$ $x_s \rightarrow f_s(x_1, x_2,... ,x_s)$.For $H\in{}^sS^r(K^*)$ and $F \in{}^sS^r(K^*)$ we define $F(H)$ as tuple $(f_1(\alpha(x_1), \alpha(x_2),... ,\alpha(x_s)), f_2(\alpha(x_1), \alpha(x_2),... ,\alpha(x_s),..., f_r(\alpha(x_1), \alpha(x_2),... ,\alpha(x_s))$for $\alpha=\alpha(H)$.

Let us consider a to totality ${}^{s}BS_{r}(K^*)$ of sequences of kind $u=(H_0, G_1, G_2, H_3, H_4, G_5, G_6, \ldots, H_{t-1}, H_t)$, $t=4i$, where $H_k \in S(K^s)$, $G_j \in S^{s,r}(K)$. We refer to ${}^{s}BS_{r}(K^*)$ as a totality of bigraded multiplicative symbolic strings.

We define a product of u with $u'=(H'_0, G'_1, G'_2, H'_3, H'_4, G'_5, G'_6, \ldots, H'_{l-1}, H_l)$ as $w=(H_0, G_1, G_2, H_3, H_4, G_5, G_6, \ldots, H_{t-1}, H'_0(H_t), G'_1(H_t), G'_2(H_t), H'_3(H_t), H'_4(H_t), G'_5(H_t), G'_6(H_t), \ldots, H'_{l-1}(H_t), H'_l(H_t))$.This operation converts ${}^{s}BS_{r}(K^*)$ into a semigroup. If H_t is an element of${}^{s}EG(K)$ then $rev(u)=(H_{t-1}(H_t^{-1}),, G_{t-2}(H_t^{-1}), G_{t-3}(H_t^{-1})), H_{t-4}(H_t^{-1}), H_{t-5}(H_t^{-1}), G_{t-6}(H_t^{-1}), G_{t-7}(H_t^{-1}),, \ldots, H_1(H_t^{-1}), H_t^{-1})$, $t=4i$, where $H_k \in S(K^s)$, Linguistic compression homomorphism ${}^{I}\psi$ *of* ${}^{s}BS_{r}(K^*)$ into ${}^{m+s}EG(K)$
can be defined for arbitrary linguistic graph $I(K^*)$ of type s,r,m via generalisation of the definition of the map given in4.2.In general case ${}^{I}\psi(rev(u)u)=e$.

Let us consider group $K'=K^*[x_1, x_2, \ldots x_s. y_1, y_2, \ldots, y_r]$ and totality ${}^{p}BS_{p}(K^*)$ for $p=s+r$ of chains of maps F of kind $x_1 \rightarrow f_1, x_2 \rightarrow f_2, \ldots, x_s \rightarrow f_s, y_1 \rightarrow g_1, y_2 \rightarrow g_2, .., y_r \rightarrow g_r$ from the semigroup ${}^{p}ES(K)$.

If r and s are chosen then we can identify F with the pair of elements ${}^{1}F=(f_1, f_2, \ldots, f_s) \in {}^{p}S^{s}(K^*)$ and${}^{2}F=(f_{1+s}, f_{2+s}, \ldots, f_p) \in {}^{p}S^{r}(K^*)$.

The product of two chains $(F_1, F_{2,\ldots,} F_{k,})$ and $(G_1, G_{2,\ldots,}, G_{t,})$ is the chain $(F_1, F_{2,\ldots,} F_{k,}, G_1(F_k), G_{2,}(F_k), {}_{\ldots,}, G_{t,}(F_k))$. Empty chain is the unity of the semigroup ${}^{p}S(K^*)$ formed by this totality of chains. In fact semigroup ${}^{p}S(K^*)$ is a semidirect product of a free semigroup over the alphabet ${}^{p}S^{p}(K^*)$ and Eulerian semigroup ${}^{p}ES(K^*)$. We refer to this object as *semigroup of strings of Eulerian transformations.*We consider also semigroup ${}^{p}RS(K^*)$ of *reversible strings* of kind $u=(F_1, F_{2,\ldots,} F_{k,})$, $F_k \in {}^{p}ES(K^*)$. For such special string we ntroduce its reverse as $rev(u)=(F_{k-1}((F_{k,})^{-1}), F_{k-2}((F_{k,})^{-1}), \ldots, F_1((F_{k,})^{-1}), (F_{k,})^{-1}))$. Let ${}^{p}S'(K^*)$ and ${}^{p}RS'(K^*)$ be subsemigroups of strings of even length in${}^{p}S(K^*)$ and ${}^{p}RS(K^*)$.

Edge (p, l) of linguistic graph $I(K^*)$, where $p \in P$, $l \in L$, pIl can be presentedvia the tuple $(p_1, p_2, \ldots., p_{s+m}, l_1, l_2, \ldots, l_r) \in (K^*)^{s+r+m}$ where $p=(p_1, p_2, \ldots., p_{s+m})$ and the tuple $(l_1, l_2, \ldots, l_r)$ is a colour of the line l.

We consider the graph $I(K^*[x_1, x_2, \ldots., x_{s+m}, y_1, y_2, \ldots, y_r])$ defined by the same list of equations with $I(K^*)$ but over larger commutative group $K'=K^*[x_1, x_2, \ldots., x_{s+m}, y_1, y_2, \ldots, y_r]$.

The following symbolic computation can be defined. Take the *symbolic edge* $x=(x_1, x_2, \ldots, x_{s+m}, y_1, y_2, \ldots, y_r)$ where x_i and y_i are generators of K' over smaller commutative group K^* and polynomial string $u={}^{p}S(K^*)$ which is a tuple $(F_1, F_2, \ldots, F_t) \in {}^{p}S(K^*)$ of strings $f_1,, f_2,, ,fp$ from $K^*[x_1, x_2, \ldots., x_{s+m}, y_1, y_2, \ldots, y_r]^p$ with even parameter t .We have to complete the following steps.

S_0. Compute the line $l=(y_1, y_2, \ldots, y_r, L_1, L_2, .., L_m)$. Noteworthy that $L_i \in K'$.

S_1. Take operation $J(l, {}^{2}F_1)$ of change the colour of l for ${}^{2}F_1$.

Let ${}^{1}l=J(l, {}^{2}F_1)$.

S_2. Compute the neighbor ${}^{1}p$ of the line of colour ${}^{1}F_1$.We have ${}^{1}p=N({}^{1}l, {}^{1}F_1)$.

S_3. Change the colour of ${}^{1}p$ for${}^{1}F_2$. *Let* ${}^{2}p=J({}^{1}p, {}^{1}F_2)$.

S_4. Compute the neighbouring line ${}^{2}l$ of ${}^{2}p$ with the colour ${}^{2}F_{2..}$

Repeat steps S_1-S_4 with initial edge ${}^{2}p$, ${}^{2}l$ and components F_3 and F_4 of the string F.

After the completion of the cycle S_1-S_4 of $d=t/2$ times we get the edge

${}^{d}p$, ${}^{d}l$ of the algorithm. Let $(P_1, P_2, \ldots, P_s, P_{s+1}, P_{s+2}, .., P_{s+m})$ coordinates ofthe line ${}^{d}p$ of the graph $I(K')$ and $L_1, L_2, \ldots, L_r$ be the colour of the line${}^{d}l$.Noteworthy that$(P_1, P_2, \ldots, P_s)= {}^{1}F_{t.}$and $(L_1, L_2, \ldots, L_r)={}^{2}F_{t.}$.

Finally we consider the map φ on edge variety $(K^*)^{s+r+m}$ of the original graph $I^*(K)$ given by the rule $x_1 \rightarrow P_1$, $x_2 \rightarrow P_2, \ldots, x_{s+m} \rightarrow P_{s+m}, y_1 \rightarrow L_1, y_1 \rightarrow L_2, \ldots, y_r \rightarrow L_r$, which is an element of ${}^{m+s+r}ES(K)$.

We refer *to* $\varphi={}^{I}\varphi$ as linguistic edge compression map of graph $I(K^*)$.

Lemma 3. *Let $I(K^*)$ be a linguistic graph of type (s, r, m) over K^* defined over multiplicative group of commutative ring K***.** *Then edge compression map $\varphi={}^{I}\varphi$: ${}^{r+s}S(K^*) \rightarrow {}^{r+s+m}ES(K^*)$ is a homomorphism of semigroups.*

Lemma 4.*Let $u \in {}^{P}RS(K^*)$ then $urev(u)$ is an element of kernel ofn${}^{I}\varphi$* **,**

Corrolary.${}^{I}\varphi({}^{r+s}RS(K^*))$ *is a subgroup of* ${}^{I}\varphi({}^{r+s}S(K^*))$.

We refer to elements *of*${}^{I}\psi({}^{s}BS_r(K^*))$and ${}^{I}\varphi({}^{r+s}S(K^*))$ as *chain transitions ofpoints*and *edges*of type(s, r, m)on the varieties$(K^*)^{r+s}$and$(K^*)^{r+s+m}$respectively.

We consider totalities ${}^{s}R^{r}$of $R^{s,r}$reversible strings from ${}^{s}BS_r(K^*)$and${}^{r+s}S(K^*)$ with last component from ${}^{s}EG'(K)$ and ${}^{s+r}EG'(K)$and ${}^{I}\psi({}^{s}R^{r})$ and ${}^{I}\varphi(R^{s,r}))$. Let ${}^{n}X(K^*)$be the totality of chain transition from sets${}^{I}\psi({}^{s}R^{r})$for all possible linguistic graphs $I(K^*)$ of type $s, r, n\text{-}s$, $0<r, s<n$ and ${}^{n}Y(K^*)$ be the totality of chain transitions from ${}^{I}\varphi(R^{s,r})$ of type $s, r, n\text{-}s\text{-}r$. We consider multiplicative linguistic group ${}^{n}LG(K^*)$ generated by elements ${}^{n}X(K^*)$,${}^{n}Y(K^*)$and all generalized Jordan-Gauss elements of ${}^{n}EG(K^*)$.

In some cases of special commutative rings K one can prove that ${}^{n}EG(K^*)={}^{n}LG(K^*)$. The following natural algorithm for generation of pair g and g^{-1} consists of four steps $S_1 - S_4$.

$S_{1.}$ take several generalised Jordan-Gauss elements $j_1, j_2, \ldots, j_k$ and compute their inverses. S_2. select pairs $s(i), r(i)$ for$i=1, 2, \ldots, t$ and corresponding linguistic graphs $L(i)=L(r(i), s(i))(K^*)$of type $s(i), r(i), n\text{-}s(i)$.Take strings $u(i)$ from the subset${}^{s(i)}R^{r(i)}$of ${}^{s(i)}BS_{r(i)}(K^*)$.Compute $rev(u)$. Take linguistic compression homomorphism ${}^{L(i)}\psi$ and compute $a_i={}^{L(i)}\psi(u(i))$ and their inverses $a_i^{-1}={}^{L(i)}\psi(rev(u(i)))$.

S_3. select pairs $s(i), r(i)$ for $i=t+1, t+ 2, \ldots, t+d$ and corresponding linguistic graphs $L(i)=L(r(i), s(i))(K^*)$. Take strings $u(i)$ from the subset $R^{s(i),r(i)}$ of ${}^{s(i)+r(i)}RS(K^*))$.Compute $rev(u)$. Take linguistic compression homomorphism ${}^{L(i)}\psi$ andcomputes $a_i={}^{L(i)}\varphi(u(i))$ and their inverses $a_i^{-1}={}^{L(i)}\varphi(rev(u(i)))$.

S_4.take alphabet $A =\{j_1, j_2, \ldots, j_{k,} ,a_1,a_{2, \ldots,} a_{t+d}\}$ and write a word g in this alphabet $z_1 z_2 \ldots z_l$ where $z_i \in A$. Then $g^{-1} =z_l^{-1} z_{l-1}^{-1} \ldots z_1^{-1}$.

5. Inplementation of algorithm 2. 1. 2 with subsemigroups *ED(n, K*)* and *EA(n, K*)* and corresponding cryptosystems.

5.1. Implementation of protocol 2.3.2.

Recall that Alice and Bob have to use algorithm 2.2 with collision map u on $(K^*)^m$ as leading procedure. So Alice works with objects related to graph $D(n, K^*)$.She takes strings$u_1, u_2, \ldots, u_s$, s>1 of $BS(K^*)$. She computes images g_i and h_i of linguistic compresision maps ${}^{D(n,K^*)}\psi$ of$BS(K^*)$ onto $ED(n, K^*)$and ${}^{A(m,K^*)}\psi$ of $BS(K^*)$ onto $EA(m, K\}$. Alice will use homomorphism φ of $ED(n, K^*)$ onto $EA(m(n), K^*)$ induced by graph homomorphism of $D(n, K^*)$ onto $A(m. K^*)$ (see section 3). Noteworthy that $\varphi(g_i)=h_i$. She use algorithm of section 4.3 and generate pairs g, g^{-1} from ${}^nLG(K)$ and $h, h^{-1}\in {}^mLG(K)$.Finally Alice computes pairs$a_i=gg_ig^{-1}$and $b_i=hh_ih^{-1}$and sends them to Bob. Further steps of algorithms follows to general scheme. As output correspondentsget collision element u from ${}^mES(K^*)$.

5.2. Conversion to a cryptosystem.

Alice uses algorithm 4.3 to generate new pair of mutually invertible elements f and f^{1}. Assume that f is given by tuple $(f_1, f_2, \ldots f_m)$ from the $K^*[x_1, x_2, \ldots, x_m]$ and u is presented by $(u_1, u_2, \ldots u_m)$.Alice computes string $(f_1u_1, f_2 u_2, \ldots, f_ku_k)$. and sends it to Bob. He restores the string $(f_1, f_2, \ldots f_m)$ and uses this map for the encryption. Alice decrypts withf^{1}.

5.3. Asymmetric schemes of multivariate cryptography on safe Eulerian mode.

Let F,F^{-1}be an asymmetric multivariate encryption scheme like one of various modifications of Imai Matsumoto MIC cryptosystem or another known bijective quadratic multivariate scheme. Assume that multivariate encryption rule F is given in its standard form. Note that procedure of computation of F^{-1} in the given point can be given as numerical algorithms. Alice selects g from ${}^{,m}LG(K)$ given by the rule $(g_1, g_2, \ldots g_m)$ and computesg^{-1}. She sends ''deformed g'' (see [16] and examples in [41]) in the form of tuple $(g_1u(f)_1, g_2u(f)_2, \ldots, g_mu(f)_m)$ together with $F(g^{-1})$ in its standard form. Bob is notified on the form of ''deformation rule''. So he restores themap F.

Correspondents works with the plainspace$(K^*)^m$ and cipherspaceK^m. Bob writes his massage p, transforms it to $p'=f(p)$ and creates the ciphertext as $F(p')=c$. Alice computes $F^{-1}(c)=c'$ and restores the plaintext as$f^{1}(c')$.

Adversary is not able to apply known methods of Algebraic Cryptology, because of encryption multivariate map $G=F(f)$ is not a bijective transformation of K^m, it has unbounded degree. Task of finding of G' on K^m such that $G(G')$ acts on $(K^*)^m$ as identity is unfeasible task because of standard form for G' is not a rule of polynomial density.

Supporting procedure is algorithm of kind 2.1 with the same commutative ring $K$and parameter m.Alice creates elements z and z^{-1} *of* ${}^mLG(K)$. She takes z of kind $x_i \to z_i(x_1, x_2, \ldots, x_m)$, $i=1,2,\ldots,m$ and forms the tuple $(z_1u_1, z_2u_2, \ldots, z_mu_m)$ to send it to Bob. He uses his knowledge on u to compute z.

Alice sets pairs (a_i, b_i) to start supporting protocol 2.1. She sends $b_i(z^{-1})$ which has polynomial density to Bob. Bob use his knowledge on z and computes b_i. Correspondents execute protocol 2.1 and get collision stable map u.

Alice uses platform of 2.1 to generate mutually invertible transformations y and y^{-1}acting on K^m. She keeps y^{-1} for herself and sends $y+u$ to Bob. He subtracts u and gets y. As in previous algorithm Alice and Bob use plainspace $(K^*)^m$ and ciphertext K^m.

To encrypt Alice maps her message p in the alphabet K^* to $z^{-1}(p)=m$ and then she computes theciphertext $c=y^{-1}(m)$. Bob decrypts via application of y *to* c and computation $z^{-1}(y(c))$.

Similarly Bob encrypts p via consecutive computation *of* z *to* p and $y(z(p))$.Alice applies y^{-1} to ciphertext c and computes the plaintext as $z^{-1}(y^{-1}(c))$.

6. Groups *GD(n, K)* and *GA(m, K)* and corresponding cryptosystems.

6.1. Implementation of algorithm 2.1.1 with groups *GD(n, K)* and *GA(m, K).*

Implementation of 2.1.2 on the base of platform $GD(n, K)$ and homomorphism of this group onto transformation group $GA(m, K)$ is very similar to the case of the inverse Tahoma protocol presented in [14]. The difference is that the outcome of directed protocol is a collision element u from $GA(m, K)$, recall that $u$is a cubic map.
Let us describe the directed protocol.

Alice takes strings $u_1, u_2, \ldots, u_l.\ l>1$ from the semigroup $\sum(K)$.She takes elements g and g' from$\sum(K)$ together with reversing strings $rev(g)$ and$rev(g')$. Alice formselements $v_i=gu_i\, rev(g)$ and $v'_i=g'u_i rev(g')$.

She takes homomorphism ${}^{\Gamma}\eta$ defined in section 3.2 for cases $\Gamma=D(n, K)$ and $A(m, K)$ and computes $y_i={}^{D(n,\ K)}\eta(v_i)$ and $z_i={}^{A(n,\ K)}\eta(v'_i)$. Alice takes affine transformations $T_1$and T_2 of free modules K^n and K^m respectively and forms cubic transformations $a_i=T_1y_i\, T_1^{-1}$ and $b_i=T_{2i}z_iT_2^{-1}$. She sends pairs $(a_i\, ,\, b_i), i=1,2,\ldots, l$ to Bob.

He takes abstract alphabet $c_1, c_2,\ldots, c_l$ and writes word $w=w(c_1, c_2, \ldots, c_l)$ of some length t, $t>l$. Bob specialize c_i as a_i and computes cubical transformation$w(a_1, a_2,\ldots, a_l)=v$ to Alice but keep specialisation $u= w(b_1, b_2, \ldots, b_l)$ for himself. Alice restores u via following steps.

S_1. Computation of $T_1^{-1}vT_1=v'$, $rev(g)v'g=v''$.

S_2. Computation of $ind(\delta)(v'')=y$.

S_3. Computation of $y'=(g')yrev(g')$ and u as $T_{2i}\mathrm{y}T_2^{-1}$.

6.2. Conversion to cryptosystem.

Alice can take two other invertible affine transformations T'_1 and T'_2 of free module K^m and generate pair of mutually inverse elements g and g^{-1} from $GA(m, K)$ and sends $h=T'_1gT'_2+u$ to Bob.

He restores encryption map $f=T'_1gT'_2$. Alice can decrypt with $T_2^{-1}g^{-1}T_1^{-1}$. The disadvantage of this cryptosystem is the fact that decryption map is also cubical one.It means that in the case of $O(n^3)$ interceptions of plaintext-ciphertext pairs the adversary is able to conduct linearization attack in time $O(n^{10})$.

Natural recommendation is to execute just $O(n^2)$ exchanges and set the new encryption rule (possibly with new session of protocol 6.1).

6. 3. Transform to Eulerian mode.

Alice can use algorithm 4. 3 for generation of z, z^{-1} from can send $zf^{1}$to Bob. He restores z.So correspondents works with plainspace $(K^{*})^{m}$ and cipherspace K^{m}. Bob encrypts his plaintext p as $c=f(z(p)$.Alice restores p as $z^{-1}f^{1}(c)$.

6.4. On schemes of quadratic multivariate cryptography on safe Eulerian mode.

Assume that scheme F , F^{-1} as in 5.3 where F is quadratic multivariate map is chosen by Alice. Let D be the differential operator $d/dx_1+d/dx_2+\ldots,+d/dx_m$. After the completion of 6.1. Alice takes the collision map u: $x_i \rightarrow u_i$ and forms the tuple $v=(Du_1, Du_{2,\ldots,} Du_m)$.Now she transforms $F=(f_1, f_2, \ldots, f_m)$ to $W=(f_{1+} v_1, f_{2+}v_2, \ldots, f_{m+}v_m)$. Alice sends W to Bob. He restores F.

Now correspondents can work on Eulerian mode.Bob transforms his plaintext p $\in(K^{*})^{m}$into $p'=z(p)$ and compute the ciphertext as $F(p')$. Alice uses computational procedure for $F^{-1}$and z^{-1} to decrypt.

6.5. On the usage of toric and stable platforms in tandem.

6.5.1 Public key algorithm with Eulerian transformations on private mode.

Correspondents can implement schemes 2.2.3 and 2.2.1 with the platforms of Sections 5 and 6. The output for each of these versions will be the collision map $u\in{}^{m}LS(K)$ and another collision element $y \in K^{m}$.

Alice can generate a public key map suggested in the paper [9] (case of arithmetical ring $Z_{,d}$ d>2) and [10] (the case of finite field).

So she generate maps z and z^{-1} from ${}^{m}LG(K)$ as in *6.3* and cubical map $f=T'_1gT'_2$ as in 6.2 and its reverse f^{1}. Alice takes composition $f(z)$ as in 6.2. She computes $f(z)+y(u)$ and sends it to Bob. He restores $f(z)$ and uses this map for encryption. Alice decrypt the ciphertext via consequtive applications of f^{1} and z^{-1} to ciphertext. Let us parameter t' stands for the length of reimage of g in $\sum(K)$.

We refer to t' as the length of the string. Computer simulations demonstrates the ''condensed matters physics'' digital effect. If t' is ''sufficiently large'', then $M(g, m ,t')$ is independent from t' constant. We have written a program for the implementation of the protocol. It written in C++ and compiled with the gcc compiler. We used an average PC with processor Pentium 3.00 GHz, 2GB memory RAM and system Windows 7. We have implemented three cases:

(1) T'_1 and T'_2 are identities ,

(2) T'_i *,i=1,2 is the* map of kind $x_1 \rightarrow x_1+ a_2x_2+\underline{a_3x_3}+ \ldots +a_mx_m$, $x_2 \rightarrow x_2$, $x_3 \rightarrow x_3$, $\ldots$, $x_m \rightarrow x_m$, $a_i \neq 0$, $i=1,2,\ldots,m$,

(3) $T_i= A_ix+ b_i$; where the majority of entries of each matrix A_i and coordinates of vector b_i are nonzero elements.

The number of monomials depends from parameters m and t' and the form of transformation T_i. Let us assume that parameter m , matrices T_i and commutative ring are chosen, So the value of $M(g, m, t')$ depends only from variable t'

Computer simulation shows that if t' is “sufficiently large” then $M(g,m, t')$ is a constant.

Table 1 . Number of monomial terms of the encryption map, $K= F_q$, $q=2^{32}$**, the case I**

	length of the string t'			
m	**16**	**32**	**64**	**128**
16	250	250	250	250
32	770	1010	1010	1010
64	1810	3074	4066	4066
128	3890	7202	12290	16322

Table 2. Number of monomial terms of the encryption map, $K= F_q$, $q=2^{32}$**, the case II**

	length of the string t'			
n	**16**	**32**	**64**	**128**
16	3426	3426	3426	3426
32	19392	26310	26310	26310
64	89472	148420	206222	206222
128	383232	692676	1161356	1633054

Table 3. Number of monomial terms of the encryption map, $K= F_q$, $q=2^{32}$**, the case III**

	length of the string			
m	**16**	**32**	**64**	**128**
16	6544	6544	6544	6544
32	50720	50720	50720	50720
64	399424	399424	399424	399424
128	3170432	3170432	3170432	3170432

Table 4. Number of monomial terms of the encryption map, $K= Z_q$, $q=2^{32}$**, the case I**

	length of the string ***t'***				
M	16	32	64	128	256
16	257	257	257	257	257
32	785	1025	1025	1025	1025
64	1841	3105	3849	4097	4097
128	3953	7265	9578	13681	15992

Table 5: Number of monomial terms of the encryption map. $K= Z_q$, $q=2^{32}$, case II

	length of the string t'				
m	16	32	64	128	256
16	8840	8840	8840	8840	8840
32	94087	113168	113168	113168	113168
64	868811	1309887	1549061	1599519	1599520
128	7380170	986066	17063985	19410643	22875533

Table 6: Number of monomial terms of the encryption map. $K= Z_q$, $q=2^{32}$, case III

	length of the word				
n	16	32	64	128	256
16	15504	15504	15504	15504	15504
32	209440	209440	209440	209440	209440
64	3065920	3065920	3065920	3065920	3065920
128	46866560	46866560	46866560	46866560	46866560

6.5.2. Correspondents can implement schemes 2.2.3 and 2.2.1 with the platforms of Sections 5 and 6.The output for each of these versions will be the collision map u $\in{}^{m}LS(K)$ and another collision element$y \in K^m$.

In this case Alice can select arbitrary element z given by a string $(z_1,z_2, ...,z_m)$ from ${}^{m}LG(K)$ and cubic (or quadratic) multivariate scheme of kind (F, F^{-1}).

She sends tuples $(z_1u_1,z_2u_2,\ldots,z_mu_m)$ and $(f_1+y_1 , f_2+y_2 , ..., f_m+y_m)$ (in the case of $deg(F)=2$ we compute $((f_1, f_2, ..., f_m) +(Dy_1,Dy_2, ...,Dy_m))$.Bob restores z and F and correspondents work with plainspace $(K^*)^m$ and K^m similarly to previous case 6.5.

6.5.3.Usage of recurrent and governing rules to work with combine multivariate transformations of different nature.

Let us assume that Alice takes several bijective transformations $F_1, F_2, ..., F_k$of degree at most 2. She can use transformation $y= (y_1, y_2, ..., y_m)$and deliver several elements ^{i}y, $i=1, 2, ..., t$ from the stable platform via recurrent procedure. One of the options is the following. Alice sends $r_1={}^{1}y+y$, $r_2 ={}^{2}y({}^{1}y)+{}^{1}y,\ldots, r_k={}^{k}y({}^{k-1}y)+{}^{k-1}y$toBob. So he computes ${}^{i}y$. Secondly she computes $D({}^{i}y)=(D({}^{i}y_1), D({}^{i}y_2), \ldots, , D({}^{i}y_m))$ and sends to Bob elements $G_i=F_i+D({}^{i}y)$ where + is an operation in $K[x_1, x_2,..., x_m]^m$.

So Bob can use a sequence of elements $u(1)={}^{1}y$, $u(2)=y^2$, $..,u(k)={}^{k}y,u(k+1)=F_1,$, $u(k+2)=F_2, ...,u(2k)=F_k$ of the alphabet A .

Alice writes *governing rule* in the form of word $w=w(z(1), z(2), ..., z(2k))=$

$z(i_1)z(i_2)...z(i_l)$ in formal alphabet Z formed by $z(i), i=1,2,...,2k$ where$i_1, i_2, ..., i_l$ is a sequence of elements from $\{1, 2, ..., 2k\}$.She sends w via open channel to Bob. He specialises $z(i_j)$ as $u(i_j)$, $j=1,2,....,2k$, writes his message as $p=(p_1, p_2, ..., p_m)$ and computes ciphertext with the procedure $c_1=u(i_1)(p),c_j=u(i_j(c_{j-1})$, $j=2,3,...,2k,c=c_{2t}$.

Alice writes reverse word and takes sequence $u(i_{2t})^{-1}$, $u(i_{2t-1})^{-1}$, ... , $u(i_1)^{-1}$ for the decryption.

Correspondents can use the above platform in tandem with the standard platform ${}^{m}LS(K)$ of toric directed Tahoma protocol with the output $u\in{}^{m}LS(K)$.Alice can generate pairs ${}^{i}u$, ${}^{i}u^{-1}$,$i=1,2,...,l$ from ${}^{m}LG(K)$. She uses open recurrent rules to compute $({}^{1}u_1{\cdot}u_1, {}^{1}u_2{\cdot}u_2, ,..., {}^{1}u_m{\cdot}u_m)={}^{1}h$, $({}^{2}u({}^{1}u)_1{\cdot}{}^{1}u_1, {}^{2}u({}^{1}u)_2{\cdot}{}^{1}u_2,..., {}^{2}u({}^{1}u)_m{\cdot}{}^{1}u_m)={}^{2}h,...,$ $({}^{l}u({}^{l-1}u)_1{\cdot}{}^{l-1}u_1, ({}^{l}u({}^{l-1}u)_2{\cdot}{}^{l-1}u_2,..., ({}^{l}u({}^{l-1}u)_m{\cdot}{}^{l-1}u_m)={}^{l}h$ for Bob.He restores ${}^{i}u=v(i)$.

Alice writes second *governing rule* in the form of word $w'=w'(z(1), z(2), ..., z(l))=$ $z(i_1)z(i_2)...z(i_t)$, $t>l-1$ in formal alphabet Z' formed by $z(i)$, $i=1,2,...,l$ wherei_1, i_2, ..., i_t is a sequence of elements from $\{1, 2, ..., l\}$.She sends w'via open channel to Bob. He specialises $z(i_j)$ as $v(i_j)$, $j=1,...., t$. writes his message as $p=(p_1, p_2, ..., p_m)$ and applies elements $v(i_1), v(i_2),..., v(i_t), u(i_{12t}), u(i_1), ..., u(i_{2k})$.

7. **Conclusion.**

Let us consider totality $V(K)$ of elements F of Cremona semigroup of polynomial degree $O(n^t)$ and polynomial density $O(n^d)$such that the restriction F' of F onto $(K^*)^m$ is an injective map and there is a polynomial algorithm of computation of reimage of element from $Im(F')=F((K^*)^m)$. We assume that element of $V(K)$ is given via its standard form. In fact we are interested only in the usage of F'. It means that we can substitute each syllable x_1^a of each monomial term for $x_1^a \bmod (K^*)$. So without loss of generality we may assume that $t=1$.

We assume that commutative ring K with unity has nontrivial multiplicative group K^*. Noteworthy that variety ${}^{m}V(K)$ contains all bijective maps of $C(K^m)$ of bounded degree for which a polynomial procedure to compute reimage x of $F(x)$ is available. Wide class of such maps is formed by explicit constructions of Multivariate cryptography designed as potential candidates for a secure public keys or stream ciphers of multivariate nature. For us existence of effective cryptanalysis for such candidates is immaterial.

Some examples of non-bijective elements of ${}^{m}V(K)$for special rings are given in [30] or [31].

Construction of group ${}^{m}LG(K)$allows to generate pair of mutually inverse elements z, $z^{-1}$of the group and to transfer selected F from ${}^{m}V(K)$ into newmap $x\rightarrow Y=F(z(x))$ from ${}^{m}V(K)$. Really both F' and $F'(z)$ have degree $O(n)$.

(2) So the owner of the pair (Alice) can announce Y written in standard form as new public key cryptosystem with the plainspace $(K^*)^m$ and ciphertext K^m.

(3) Alternatively Alice and her correspondent (Bob) can use cryptosystem of El Gamal type based on subsemigroups of ${}^{n}ES(K)$ and ${}^{m}ES(K)$ (see [32]). Security of this cryptosystem is based on the word problem. Notice that together of algorithm of the section 4.3 inverse protocol can be used in the wide case of finite commutative ring with nontrivial multiplicative group. So correspondents elaborate pair u, u^{-1} where u belongs to Alice and u^{-1} is in the possession of Bob.

Alice send $F(z(u))$ to Bob and he restores $Y=F(z)$. Bob can write plaintext $p\in(K^*)^m$ and form ciphertext as $Y(p)$. Alice can compute $c'=F^{-1}(c)$ and compute his plaintext as $z^{-1}(c')$.

Notice that this algorithm is asymmetrical. Bob does not have '' local inverse''Y' of Y for which $Y'Y$ acts identically on the variety $(K^*)^m$.

(4) For safe delivery of Y to Bob correspondents may use direct Tahoma protocol with two platforms $^{n}ES(K)$ and $GD(n,K)$. So they elaborate $u\epsilon\ ^{m}ES(K)$ and $g\ \epsilon\ GD(m.K)$ for Alice and Bob. Alice sends $ug+F(z)$ to Bob. He restores $F(z)$ via subtraction of ug.The remaining part of such algorithm is same with previous one.

Correspondents can use symmetric scheme because Alice can deliver z and F on secure mode via schemes of section 6.

Known methods of algebraic cryptanalysis with the usage of Shirshov-Grobner algorithms are not applicable to suggested above cryptosystems especially in the cases of alternative form to public key cryptosystems.

References

1. Kocarev, L. J., Halle, K. S., Eckert, K., Chua, L. O. &Parlitz, U**.,** Experimental demostration of secure communications via chaos synchronization, Int. J. Bifurcation and Chaos, 1992, 2, 709–716
2. Habutsu, T., Nishio, Y., Sasase, I. & Mori, S. ,A secret key cryptosystem by iterating a chaotic map, in Eurocrypt'91 (Springer-Verlag), pp. 127–136.
3. Pecora, L. M. &Caroll, T. L. Synchronization in chaotic systems, Phys. Rev. Lett. , 1990, 64, 821–824.
4. Kotulski, Z., Szczepanski, ´ J., G´orski, K., Paszkiewicz, A. &Zugaj, A. Application of discrete chaotic dynamical systems in cryptography–DCC method, Int. J. Bifurcation and Chaos, 1999, 9, 1121–1135.
5. LjupcoKocarev&ShiguoLian Springer, Chaos-Based Cryptography, Springer, 2011.
6. V. Ustimenko, Linguistic Dynamical Systems, Graphs of Large Girth and Cryptography, Journal of Mathematical Sciences, Springer, vol.140, N3 (2007) pp. 412-434.
7. 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 427/2012, 257-285.
8. T. Shaska, V. Ustimenko, On the homogeneous algebraic graphs of large girth and their applications, Linear Algebra and its Applications Article, Volume 430, Issue 7, 1 April 2009, Special Issue in Honor of Thomas J. Laffey.
9. V. Ustimenko, On new multivariate cryptosystems based on hidden Eulerian equations, Reports of Nath . Acad of Sci, Ukraine, 2017. № 5, pp 17-24.
10. V. Ustimenko, On new multivariate cryptosystems based on hidden Eulerian equations over finite fields, ePrint Archive, 093, 2017.
11. Post-Quantum Cryptography: Call for Proposals:https://csrc.nist.gov/Project; Post-Quantum-Cryptography-Standardization/Call-for-Proposals, Post-Quantum Cryptography: Round 2 Submissions
12. M. Andrzejczak, The Low –Area FPGA Desighn for the Post – Quantum Cryptography Proposal Round 5, Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS), Cryptography and Security Systems, Leipzig, September, 2019 (to appear).
13. R. J. McEliece, A Public-Key Cryptosystem Based On Algebraic Coding Theory (1978), DSN Progress Report, 44: 114–116.
14. V.Ustimenko, On inverse protocols of Post Quantum Cryptography based on pairs of noncommutative multivariate platforms used in tandem, ePrint Archive, 897, 2019.

15. R. Wagner, M. R. Magyarik, ``A Public-Key Cryptosystem Based on the Word Problem", Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA, August 19-22, 1984.
16. 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.
17. 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.
18. DelaramKahrobaei and 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.
19. Alexei Myasnikov; Vladimir Shpilrain and Alexander Ushakov (2008), Group-based Cryptography, Berlin: BirkhäuserVerlag.
20. Zhenfu Cao (2012), New Directions of Modern Cryptography. Boca Raton: CRC Press, Taylor & Francis Group. ISBN 978-1-4665-0140-9.
21. Alexei G. Myasnikov; Vladimir Shpilrain and Alexander Ushakov, Non-commutative Cryptography and Complexity of Group-theoretic Problems, American Mathematical Society, 2011..
22. I. Anshel, M. Anshel and D. Goldfeld, An algebraic method for public-key cryptography. Math. Res.Lett. 6(3–4), 287–291 (1999).
23. S.R. Blackburn and S.D. Galbraith, Cryptanalysis of two cryptosystems based on group actions. In: Advances in Cryptology—ASIACRYPT '99. Lecture Notes in Computer Science, vol. 1716, pp. 52–61. Springer, Berlin (1999).
24. K.H. Ko, S.J. Lee, J.H. Cheon, J.W. Han, J.S. Kang and C. Park, New public-key cryptosystem using braid groups. In: Advances in Cryptology—CRYPTO 2000, Santa Barbara, CA. Lecture Notes in Computer Science, vol. 1880, pp. 166–183. Springer, Berlin (2000)
25. G. Maze, C. Monico and J. Rosenthal, Public key cryptography based on semigroup actions, Adv.Math. Commun. **1**(4), 489–507 (2007).
26. P.H. Kropholler and S.J. Pride , W.A.M. Othman K.B. Wong, P.C. Wong, Properties of certain semigroups and their potential as platforms for cryptosystems, Semigroup Forum (2010) 81: 172–186.
27. J.A. Lopez Ramos, J. Rosenthal, D. Schipani and R. Schnyder, Group key management based on semigroup actions, Journal of Algebra and its applications, vol.16 (to appear in 2019).
28. 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.
29. V. Ustimenko, U. Romanczuk-Polubiec, A. Wroblewska, M. Polak, E. Zhupa, On the constructions of new symmetric ciphers based on non-bijective multivariate maps of prescribed degree,Security and Communication Networks, Volume 2019, Article ID 2137561, 15pages https://doi.org/10.1155/2019/2137561.
30. V. A. Ustimenko, "On Schubert cells in Grassmanians and new algorithms of multivariate cryptography", *Tr. Inst. Mat.*, **23**:2 (2015), 137–148.
31. Vasyl Ustimenko, "On algebraic graph theory and non-bijective multivariate maps in cryptography", *Algebra Discrete Math.*, **20**:1 (2015), 152–170.
32. V. Ustimenko, On semigroups of multiplicative Cremona transformations and new solutions of Post Quantum Cryptography, Cryptology ePrint Archive, 133, 2019.
33. N. Biggs, Algebraic graphs theory, Second Edition, Cambridge University Press, 1993.

34. M. Polak, U. Roma´nczuk, V. Ustimenko and A. WrÃ¸sblewska, “On the applications of Extremal Graph Theory to Coding Theory and Cryptography”, Electronic Notes in Discrete Mathema Discrete Mathematics, N43, 2013, p. 329-342.
35. F. Lazebnik, V. Ustimenko and A. J.Woldar, “A new series of densegraphs of high girth”, *Bull. Amer. Math. Soc.* (N.S.) 32, no. 1, 1995, pp. 73-79
36. V. Ustimenko, “On extremal graph theory and symbolic computations”,Dopovidi National Academy of Sciences of Ukraine, , N2, 2013, pp. 42-
37. V. Ustimenko,U. Roma´nczuk-Polubiec, A. Wroblewska, Expanding graphs of the Extremal Graph Theory and expanded platforms of Post QuantumCryptography, Proceedings of Federated Conference on Computer Science and Information Systems (FedCSIS), 2019, Leipzig, September (to appear).
38. V. Ustimenko, “Maximality of affine group, hidden graph cryptosystemand graph’s stream ciphers”, Journal of Algebra and Discrete Mathematics, 2004, v.10, pp. 51-65.
39. 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 ofAdvances in Intelligent Systems and Computing (AISC, volume 998), pp, 654-674.
40. V. Ustimenko, M. Klisowski, On Noncommutative Cryptography and homomorphism of stable cubical multivariate transformation groups of infinite dimensional affine spaces,Cryptology ePrint Archive, 593, 2019.
41. 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.
42. Max Noether, Luigi Cremona, MathematischeAnnalen 59, 1904, p. 1–19.