On inverse protocols of Post Quantum Cryptography based on pairs of noncommutative multivariate platforms used in tandem.

V. Ustimenko

University of Maria Curie Sklodowska in Lublin, Poland
vasyl@hektor.umcs.lublin.pl

Abstract.
Non-commutative cryptography studies cryptographic primitives and systems which are based on algebraic structures like groups, semigroups and noncommutative rings. We continue to investigate inverse protocols of Non-commutative cryptography defined in terms of subsemigroups of Affine Cremona Semigroups over finite fields or arithmetic rings Z_m and homomorphic images of these semigroups as possible instruments of Post Quantum Cryptography. This approach allows to construct cryptosystems which are not public keys, as outputs of the protocol correspondents receive mutually inverse transformations on affine space K^n or variety $(K^*)^n$ where K is a field or an arithmetic ring.

The security of such inverse 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. We discuss the idea of the usage of combinations of two cryptosystems with cipherspaces $(K^*)^n$ and K^n to form a new cryptosystem with the plainspace $(K^*)^n$, ciphertext K^n and nonbijective highly nonlinear encryption map.

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.

Post-Quantum Cryptography (PQC) is an answer to a threat coming from a full-scale quantum computer able to execute Shor’s algorithm. With this algorithm implemented on a quantum computer, currently used public key schemes, such as RSA and elliptic curve cryptosystems, are no longer secure. The U.S. NIST made a step toward mitigating the risk of quantum attacks by announcing the PQC standardisation process [1]. In March 2019 NIST published a list of candidates qualified to the second round of the PQC process. Nowadays hardware performance of Round 1 candidates was reported for only a small percentage of all submissions. Few public key candidates are implemented like PQC Round 2 candidate called Round 5 (see [2]) or code based classic Mc Eliece algorithm (see [3]).

In this publication we continue to develop new cryptosystems within alternative approach ([4], [5], [6]) to public key cryptography based on the idea of modified Diffie Hellman type protocol which output is a pair of mutually inverse multivariate transformations of affine space K^n defined over finite commutative ring K. Security of these algorithms rests on the complexity of word problem to decompose given multivariate map into generators of affine Cremona [7] semigroup. The first usage of the complexity of word problem for groups was considered in [8].

In the algorithms of this paper the encryption rule is not given publicly. We introduce new cryptosytems defined in terms of stable semigroups of transformations of affine K^n which consist of transformations of degree bounded by small constant. Main instruments are the following. Let K be a commutative ring, $K[x_1, x_2, \ldots, x_n]$ be a ring of polynomials in *n variable. Semigroup of endomorphisms* $End(K[x_1, x_2, \ldots, x_n])=S(K^n)$ of $K[x_1, x_2, \ldots, x_n]$ is known as Affine Cremona Semigroup , element f of $S(K^n)$ acts naturally on affine space K^n and can be given its standard form $x_1 \to f_1(x_1, x_2, \ldots, x_n)$, $x_2 \to f_2(x_1, x_2, \ldots, x_n)$, … $x_n \to f_n(x_1, x_2, \ldots, x_n)$, where $f_1 \epsilon K[x_1, x_2, \ldots, x_n]$.

We assume that K is a finite commutative ring. Symbol $C(K^n)$ stands for Affine Cremona Group of all invertible elements from $S(K^n)$.

Density of the map f is the total number of monomial terms in all f_i. We say that $f \epsilon C(K^n)$ is computationally tame if densities of f and f^{-1} are of sizes $O(n^d)$ and $O(n^t)$ for some constants d and t. Let us consider illustrative examples of the usage of these objects in Cryptography.

1.1. Commutative case, group based inverse Diffie-Hellman protocol.

Alice generates pair g and g^{-1} from the subgroup G of $C(K^n)$. Correspondents work with cyclic group $<g>$ of large order. Alice computes $h=g^{\alpha}$ where α is her positive key. She sends h to Bob together with g^{-1}. Bob computes $f=(g^{\alpha})^{\beta}$ with his key β and sends $(g^{-1})^{\beta}$ to Alice. She computes f^{-1} as $(g^{-\beta})^{\alpha}$.

So Alice and Bob can use free module K^n as plainspace, maps f^{-1} and f- as encryption tools. They can decrypt via application of their f^{-1} and f .

The implementation of this scheme is computationally heavy because for g and h in ‘’general position degree’’ of $g(h(x))$ coincides with degree of $h(g(x))$ and equals $deg(g) \cdot deg(h)$. The density of g^x is growing fast when x grows.

We know two conditions

(1) *stability condition*, group G such that for each $g \epsilon G$ maximal degree $deg(g)$ is d (the cases $d=2$ or $d=3$ are probably the most important).

(2) *minimality of density condition* (transformation $g \epsilon G$ has to be toric, i.e. its standard form is written as $x_i \to t_i(x_1, x_2, \ldots, x_n)$, where t_i are monomial expressions. We refer to g as Eulerian map if coefficients are regular coefficients and the map g is bijective one on the variety $(K^*)^n$. Correspondents use this variety as the plainspace. Let ${}^nEG(K)$ be Eulerian group of all such transformations.

PLATFORMS. We discover classes of subgroups of kind (1) or (2) and fast algorithm to generate pairs g and g^{-1}. Look at cryptology e-print archive papers [9] and [6] and further references.

SECURITY rests on the complexity of DISCRETE LOGARITHM PROBLEM (D.L.P.) FOR GROUP G. The complexity heavily depends on G and the way group data is given (F^*_q breakable by Quantum Computer, D.L.P. for various elliptic curves, in case of $C(K^n)$ depends on the choice of the generator).

1.2. Commutative case, inverse protocol in the case of semigroups.

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

Alice takes *f* and *h* from *S'* such that $\varphi(fh)=1$. Let $g=\varphi(f)$. She computes g^{α} where α is positive integer and sends it to Bob together with *h*.

Bob computes $z=(g^{\alpha})^{\beta}$ where β-positive integer together with $y=h^{\beta}$. He sends *y* to Alice.

She compute $\varphi(y^{\alpha})$ which coincides with z^{-1}.

Then they can use plainspace K^m and z and z^{-1} as encryption and decryption function.

ADVERSARY has to solve DISCRETE LOGARITHM PROBLEM FOR SEMIGROUP *S'* (solve $h^x=y$).

Recall that subsemigroup *S'* and group *G* have to be *stable* , i. e. degree of elements bounded by *d (d=2, 3)* or *S'* is a subsemigroup of *toric maps* on K^n and $G <^m EG(K)$.

PLATFORMS (pairs *S', G* can be found in texts of cryptology e-print archive [32], [6] (see further references). In the case of finite fields platforms are defined in terms of algebraic graphs of large girth and incidence graphs of finite geometries.

1.3. Elements of Non-Commutative Cryptography with multivariate transformations.

Notice that security of Diffie-Hellman algorithm for groups depends not only on abstract group *G* but on the way of its generation in computer memory. For instance if $G=Z^*_p$ is multiplicative group of large prime field then discrete logarithm problem (DLP) is difficult one and guarantees the security of the protocol, if the same abstract group is given as additive group of Z_{p-1} protocol is insecure because DLP will be given by linear equation.

If *G* is noncommutative group correspondents can use conjugations of elements involved in protocol, some algorithms of this kind were suggested in [10], [11], [12], [13], where group *G* is given with the usage of generators and relations. Security of such algorithms is connected with Conjugacy Search Problem (CSP) and Power Conjugacy Search Problem (PCSP), which combine CSP and Discrete Logarithm Problem and their generalisations.

This direction belongs to **Non-commutative cryptography** which is an active area of cryptology, where the.cryptographic primitives and systems are based on algebraic structures like groups, semigroups and noncommutative rings (see [14], [15], [16], [17], [18], [19], [20], [23], [24]. One of the earliest applications of a non-commutative algebraic structure for cryptographic purposes was the usage of braid groups to develop cryptographic protocols. Later several other non-commutative structures like Thompson groups and Grigorchuk groups have been identified as potential candidates for cryptographic post quantum applications. The standard way of presentations of groups and semigroups is the usage of generators and relations (Combinatorial Group Theory). Semigroup based cryptography consists of general cryptographical schemes defined in terms of wide classes of semigroups and their implementations for chosen semigroup families (so called platform semigroups).

As we already mentioned we work with subsemigroups of affine Cremona semigroup $S(K^n)$ on generalisations and modifications of Diffie – Hellman protocols for the case of several generators. Elements of the subsemigroup are presented in their standard form of multivariate cryptography.

2. Some schemes of noncommitative cryptography with multivariate platforms.

2.1. Under conditions of section 1. 1 in the case of *stable subsemigroup* S, $S'<S<S(K^n)$ and stable group H, $G<H<C(K^m)$. Alice selects elements $s_1, s_2, \dots, s_r$, $r>1$ of subsemigroups S' and computes $\varphi(s_i)^{-1} = u_i$. She takes invertible elements $h\epsilon S(K^n)$ of kind av, $deg(a)=1$, $v\epsilon S$ and $f\epsilon C(K^n)$, $f=bg$, $deg(b)=1$, $g\epsilon H$ and forms pairs ($a_i=hs_ih^{-1}$, $b_i=f u_i f^{-1}$) and sends them to Bob.

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

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

So Alice and Bob when the protocol ends have mutually inverse encryption/decryption tools u^{-1} and u for the plainspace K^m.

Description of some implemetations of this algorithm can be found in [6].

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 elements $s_1, s_2, \dots, s_r$, $r>1$ of semigroups S and computes $\varphi(s_i)^{-1} = u_i$. She takes invertible elements h and f to form pairs ($a_i=hs_ih^{-1}$, $b_i=f u_i f^{-1}$) and sends them to Bob. The rest of the algorithm is identical to case of procedure 2.1.

After the completion of inverse protocol Alice and Bob have bijective maps u^{-1} and u on the plainspace $(K^*)^m$.

Security base: The adversary has to solve the ***word problem*** for the subsemigroup S', i. e., find the decomposition of w from S' into generators a_i, $i = 1, 2,\dots, t$. 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 antiisomorphism σ between semigroup $<a_1, a_2, \dots, a_r>$ and group $<b_1, b_2, \dots, b_r>$ 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 output u^{-1} and 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 (or Bob) deforms the input of 2.2 for her/his correspondent via the change of a_i, b_i for a_i, b_iv, $i=1,2,\dots, r'$ where v is u^{-1} or u. Notice that the maps b_iv are well defined injective maps of $(K^*)^m$ into K^m, they have *polynomial density*.

Bob (or Alice) computes pairs (a_i ,b_i) because of his/her possession of v^{-1}. After the completion of supporting procedure Alice and Bob get mutually inverse elements z^{-1} and z of ${}^{m}EG(K)$. They use $(K^*)^m$ *as* plainspace and K^m as cipherspace.

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

Bob decrypts via application of *u to c* and computation $z(u(c))$.

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

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

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 [25]. 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 output u^{-1} and u on $(K^*)^m$ as leading procedure. Supporting procedure is algorithm of kind 2.1 with the same commutative ring K and parameter m.

2.4. Let us consider the simplifications of 2.3.1. and 2.3.2.

Instead of supporting inverse protocols Alice generate pair of elements z and z^{-1} from ${}^{m}EG(K)$ (the case of 2.1) or pair of computationally tame elements y and y^{-1} of $C(K^m)$ (case of 2.2).

Correspondents execute procedure 2.1 (or 2.2) and Alice sends $u^{-1}(z)$ (or $u^{-1}(y)$) to Bob. He uses his map u to compute z (or y).

Alice encrypts her message p from $(K^*)^m$ via the computation $u^{-1}(z^{-1}(p))$(the case 2.1) or computation of $z^{-1}(u^{-1}(p))$ (the case 2. 2).

Bob gets the ciphertext and decrypts it as $z((u(c)))$ (case 2.1) or $u(z(c))$(case 2.2).

In his term Bob encrypts his plaintext as $u(z(p))$ case 2.1 or $z(u(p))$ case 2.2 and Alice decrypts via computation of $z^{-1}(u^{-1}(c))$ (2.1) or $u^{-1}(z^{-1}(c))$(case 2.2).

Remark. In the case 2.2 Alice (or Bob) instead of mutually invertible y , y^{-1} can use elements w , $w' \epsilon S((K)^m$ of polynomial density such that their y^{-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 [26] and [27].

Algorithms of generation of pairs (z, z^{-1}) from ${}^{m}EG(K)$ are described in [6].

3. On groups and semigroups defined in terms of linguistic graphs.

3.1. On linguistic graphs over commutative rings and skating on them.

The missing definitions of graph-theoretical concepts which appear in this paper can be found in [28]. All graphs we consider are *simple* graphs, i.e. undirected without loops and multiple edges. Let $V(G)$ and $E(G)$ denote the set of vertices and the set of edges of G respectively.

When it is convenient we shall identify G with the corresponding anti-reflexive binary relation on $V(G)$, i.e. $E(G)$ is a subset of $V(G)\circ V(G)$ and write $v\ G\ u$ for the adjacent vertices u and v (or neighbours).

We refer to $|\{x \in V(G)|\ xGv\}|$ as *degree of the vertex v.*

The *incidence structure* is the set V with partition sets *P(points)* and *L (lines)* and symmetric binary relation I such that the incidence of two elements implies that one of them is a point and another one is a line. We shall identify I with the simple graph of this incidence relation or *bipartite graph*. The pair $x,\ y,\ x \in P,\ y \in L$ such that $x\ I\ y$ is called a *flag* of incidence structure I.

Let K be a finite commutative ring. We refer to an incidence structure with a point set $P=P_{s,m}=K^{s+m}$ and a line set $L=L_{r,m}=K^{r+m}$ as linguistic incidence structure I_m if point $x=(x_1, x_2, \ldots, x_s, x_{s+1}, x_{s+2}, \ldots, x_{s+m})$ is incident to line $y=[y_1, y_2, \ldots, y_r, y_{r+1}, y_{r+2}, \ldots, y_{r+s}]$ if and only if the following relations hold

$$a_1x_{s+1}-b_1y_{r+1}=f_1(x_1, x_2, \ldots, x_s, y_1, y_2, \ldots, y_r)$$

$$a_2x_{s+2}-b_2y_{r+2}=f_2(x_1, x_2, \ldots, x_s, x_{s+1}, y_1, y_2, \ldots, y_r, y_{r+1})$$

$$\ldots$$

$$a_mx_{s+m}-b_my_{r+m}=f_m(x_1, x_2, \ldots, x_s, x_{s+1}, \ldots, x_{s+m}, y_1, y_2, \ldots, y_r, y_{r+1}, \ldots, y_{r+m})$$

where a_j, and b_j, $j=1,2,,,,m$ are not zero divisors, and f_j are multivariate polynomials with coefficients from K [29]. Brackets and parenthesis allow us to distinguish points from lines.

The colour $\rho(x)=\rho((x))$ ($\rho(y)=\rho([y])$) of point x (line [y]) is defined as projection of an element (x) (respectively [y]) from a free module on its initial s (relatively r) coordinates. As it follows from the definition of linguistic incidence structure for each vertex of incidence graph there exists unique neighbour of a chosen colour.

We refer to $\rho((x))=(x_1, x_2, \ldots, x_s)$ for $(x)=(x_1, x_2, \ldots, x_{s+m})$ and $\rho([y])=(y_1, y_2, \ldots, y_r)$ for $[y]=[y_1, y_2, \ldots, y_{r+m}]$ as the colour of the point and the colour of the line respectively. For each $b \in K^r$ and $p=(p_1, p_2, \ldots, p_{s+m})$ there is a unique neighbour of the point $[l]=N_b(p)$ with the colour b. Similarly for each $c \in K^s$ and line $l=[l_1, l_2, \ldots, l_{r+m}]$ there is a unique neighbour of the line $(p)=N_c([l])$ with the colour c. The triples of parameters s,r,m defines *type of linguistic graph.*

We consider also linguistic incidence structures defined by infinite number of equations.

Linguistic graphs are defined up to isomorphism. We refer to written above equations as canonical equations of linguistic graph.

In the case of linguistic graph defined over commutative ring the walk consisting of its vertices $v_0, v_1, v_2, \ldots, v_k$ is uniquely defined by initial vertex v_0, and colours $\rho(v_i)$, $i=1, 2, \ldots, k$ of other vertices from the path. We consider the equivalence relations on partition sets such that $(p)\approx(p')([l] \approx [l'])$ if $p_{i+s}=p'_{i+s}$ $(l_{i+r}=l'_{i+r})$ for $i \in \{1,2,..m\}$.

We define *jump operator* $J(p, a)$, $a \in K^s$ on partitions set P ($J(l,a)$, $a \in K^r$ on partion set L) by conditions $J(p,a)\approx(p)$ and $\rho(J(p,a))=a$ ($J([l],a)\approx[l]$ and $\rho(J([l],a))=a$).

Already defined *neighbour computation operator* (or *ground moving operator*) $N(v, a)$ acts on PUL by rules $N(p, a)=[l]$ where $(p)I[l]$, $\rho([l])=a$ and $N([l],a)=(p)$ where $(p)I[l]$, $\rho((p))=a$.

Let us consider *skating chain* of the linguistic graph with starting point p which is a sequence$(p, p_0, l_1, l_2, p_3, p_4, \ldots, l_{t-3}, l_{t-2}, p_{t-1}, p_t)$, $t=4k$, $k\geq 0$ such that $p\approx p_0, l_{2i+1}\approx l_{2i+2}$, $i\geq 0$, $p_{2i+1}\approx p_{2i+2}$ and $p_{2i} I l_{2i+1}$ for $i\geq 0$.

Colours of elements from the skating chain and the starting point determine the sequence. Obviously sequence of alternating jump operators J_a and ground moving operators form the skating chain from starting point (p). In fact term skating chain is selected because of the similarity of computation the sequence with competitions on skating boards, roller skates, figure skating (various jumps and skate surface moves).

3.2. Semigroups of infinite symbolic strings and linguistic compression maps.

Let us consider semigroup $S(K^s)$ and the totality $S^{s,r}(K)$ of maps of kind $G:(y_1, y_2, \ldots, y_r)\to(f_1(x_1, x_2, \ldots, x_s), f_2(x_1, x_2, \ldots, x_s), \ldots, f_r(x_1, x_2, \ldots, x_s))$. If $H\epsilon S(K^s)$ then $G(H)$ for $G\epsilon S^{s,r}(K)$ is the map $(y_1, y_2, \ldots, y_r)\to(f_1(H(x_1),H(x_2), \ldots, H(x_s)), f_2(H(x_1),H(x_2), \ldots, H(x_s)), \ldots, f_r(H(x_1),H(x_2), \ldots, H(x_s)))$.

When it is convenient we will identify elements of $S(K^s)$ with tuples from $K[x_1, x_2, \ldots, x_s]^s$ and elements of $S^{s,r}(K)$ with tuples of $K[x_1,x_2, \ldots,x_s]^r$.

Let us consider a to totality ${}^sBS_r(K)$ 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 $H_k\epsilon S(K^s)$, $G_j \epsilon S^{s,r}(K)$. We refer to ${}^sBS_r(K)$ as a totality of bigraded 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 $w=(H_0, G_1, G_2, H_3, H_4, G_5, G_6, \ldots, H_{t-1}, H'_0(H_t), G'_1(H_t), G'_2(H_t), H'_3(H_t), H'_4(H_t), G'_5(H_t), G'_6(H_t), \ldots, H'_{l-1}(H_t), H'_l(H_t))$.

It is easy to see that this operation transforms ${}^sBS_r(K)$ into the semigroup with the unity element (H_0), where E_0 is an identity transformation from $S(K^s)$.

Elements of kind $(H_0, G_1, G_2, H_3, H_4)$ are generators of the semigroup.

We refer to generator with $H_4=E_0$ as loop element. Let $L={}^sL_r(K)$ be the totality of loop elements. The semigroup generated by loop elements is isomorphic to free semigroup $F(L)={}^sF_r(K)$ of words in the alphabet L. We refer to $F(L)$ as semigroup of loop strings.

It is easy to see that ${}^sBS_r(K)$ is isomorphic to semidirect product of $F(L)$ and affine Cremona semigroup $S(K^s)$.

Let us consider the homomorphism of the group ${}^sBS_r(K)$ into Cremona Semigroup $S(K^{s+m})$ defined in terms of linguistic graph $I=I^m(K)$. Notice that one can consider graph $I^m(K')$ over the extension K' of K with the usage of the same equations.Let us take $K'=K[x_1, x_2, \ldots, x_{m+s}]$ where x_i are formal variables and consider an infinite graph $I^m(K[x_1,x_2, \ldots,x_n])$, $n=m+s$ with partition sets $P'=K[x_1,x_2, \ldots, x_{m+s}]^{m+s}$ and $L'=K[x_1, x_2, \ldots, x_{m+s}]^{m+r}$. After that we take a bipartite string $u=(H_0, G_1, G_2, H_3, H_4, G_5, G_6, \ldots, H_{t-1}, H_t)$ formed by a totality of multivariate polynomials from the subring $K[x_1, x_2, \ldots,x_s]$ of $K[x_1, x_2, \ldots,x_n]$ and the point $(x)=(x_1, x_2, \ldots, x_n)$ formed by generic elements of K'. This data defines uniquely a skating chain

$(x),J((x),H_0)=({}^1x),N(({}^1x),G_1)=[{}^2x],J([{}^2x],G_2)=[{}^3x],N([{}^3x],H_3)=({}^4x),J(({}^4x),H_4)=({}^5x), \ldots, J([{}^{t-2}x],G_{t-2})=[{}^{t-1}x],N([{}^{t-1}x],H_{t-1})=({}^tx),J(({}^tx),H_t)=({}^tx)$.

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

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

Lemma 1. *The map* $\psi={}^I\psi$: ${}^sBS_r(K)\to S(K^n)$ *is a homomorphism of semi-groups.*

We refer to ${}^I\psi({}^sBS_r(K))={}^ICT(K)$ as a *chain transitions semigroup* of linguistic graph $I(K)$ and to map ψ as *linguistic compression map.* Notice that in the case of the finite commutative ring homomorphism ψ maps infinite semigroup into finite set of chain transitions.

3.3. Some subsemigroups of symbolic strings and their homomorphic linguistic graphs over commutative rings and skating on them.

We define subsemigroup ${}^sGS_r(K)$ of *symbolic ground strings* as a totality of bipartite strings $u=(H_0, G_1, G_2, H_3, H_4, G_5, G_6,\dots, H_{t-1}, H_t)$ in ${}^sBS_r(K)$ with

$H_0=E_0,\ G_1=G_2,\ H_3=H_4, G_5=G_6,\dots,\ H_{t-1}=H_t$ and refer to ${}^I\psi({}^sGS_r(K)={}^IGCT(K)$ as *semigroup of ground chain transitions* on linguistic graph I.

Let us assume that H_t is bijective map and its inverse is a polynomial map (in the case of infinite ring K). Then we can consider a reverse bigraded string $Rev(u)= (H_{t-1}(H_t^{-1}),\ G_{t-2}(H_t^{-1}),\ G_{t-3},(H_t^{-1}),\ H_{t-4}(H_t^{-1}),H_{t-5}{}^1(H_t),\ \dots,G_2(H_t^{-1}),\ \ G_1(H_t^{-1}),\ H_0(H_t^{-1}),\ H_t^{-1})$ and refer to u as reversible string. Let ${}^sBR_r(K)$ stand for the semigroup of reversible strings.

Lemma 2. *The homomorphic image* ${}^I\psi({}^sBR_r(K))=BCT_I(K)$ *is a subgroup of affine Cremona group* $C(K^n)$.

Really ${}^I\psi(u\cdot Rev(u)),\ u\epsilon{}^sBR_r(K)$ is an identity map.

We refer to $BCT_I(K)$ as subgroup of bijective chain transitions of linguistic graph I.

3.4. Special homomorphisms of linguistic graphs and corresponding semigroups.

Let $I(K)$ be linguistic graph over commutative ring K defined in section 3.1. and $M = \{m1, m2,\dots, md\}$ be a subset of $\{1, 2, \dots, m\}$ (set of indexes for equations). Assume that equations indexed by elements from M of the following kind

$a_{m1}x_{m1} - b_{m1}y_{m1}=f_{m1}(x_1, x_2,\ \dots, x_s, y_1, y_2, \dots, y_r)$

$a_{m2}x_{m2} - b_{m2}y_{m2} = f_{m2}(x_1, x_2, \dots, x_s, x_{m1}, y_1, y_2, \dots, y_r, y_{m1})$

…

$a_{md}x_{md} - b_{md}y_{md} = f_{md}(x_1, x_2, \dots, x_s, x_{m1}, x_{m2}, \dots, x_{m\,d-1}, y_1, y_2, \dots, y_r, y_{m1}, y_{m2}, \dots, y_{m\,d-1})$

define other linguistic incidence structure I_M. Then the natural projections

$\delta_1: (x)\to(x_1, x_2, \dots, x_s, x_{m1}, x_{m2},\dots, x_{md})$ and $\delta_2: [y]\to[y_1, y_2, \dots, y_r, y_{m1}, y_{m2},\dots, y_{md}]$ of free modules define the natural homomorphism δ of incidence structure I onto I_M.

We will use the same symbol ρ for the colouring of linguistic graph I_M.

It is clear, that δ is colour preserving homomorphism of incidence structures (bipartite graphs). We refer to δ as symplectic homomorphism and graph I_M as symplectic quotient of linguistic graph I. In the case of linguistic graphs defined by infinite number of equations we may consider symplectic quotients defined by infinite subset M (see [30], where symplectic homomorphism was used for the cryptosystem construction).

Lemma 3. *A symplectic homomorphism* $\acute{\eta}$ *of linguistic graph* I *of type* (r, s, m) *onto* I' *defined over commutative ring* K *induces the semigroup homo-*

morphism $\dot{\eta}^*$ *of* $^{I}CT(K)$ *into* $^{I'}CT(K)$ *and the following diagram is commutative*

$$\begin{array}{ll} ^{s}BS_r(K) \rightarrow & ^{I}CT(K) \\ \downarrow & / \\ ^{I'}CT(K) & \end{array}$$

where horizontal and vertical arrows corresponds to linguistic compression homomorphisms $^{I}\psi$ *and* $^{I'}\psi$ *and symbol / corresponds to* η^*.

If S is a stable subsemigroup of $^{I}CT(K)$ (or $BCT_I(K)$) of degree d then $\dot{\eta}^*(S)$ is also a stable subsemigroup (or subgroup).The degree of $\dot{\eta}^*(S)$ is bounded above by d.We will search for subsemigroup X *of* $^{s}BS_r(K)$ and linguistic graphs $I(K)$ such that $\Psi(X)$ is a stable subsemigroups of $^{I}CT(K)$.

We consider more general concept of linguistic homomorphism ξ of linguistic incidence systems P, L, $I(K)$ over commutative ring K and induced by linear projections δ of P and $\delta'$of L defined via deleting of some coordinates of colour tuples $(x_1, x_1, \dots, x_s)$ and $[y_1, y_2, \dots, y_r]$ together with simultaneous deleting of x_{i+r} and y_{i+s} for i from some subset of $\{1, 2, \dots, m\}$.The image of ξ is a linguistic graph of type s_1, r_1, m_1 where $s_1 \le s$, $r_1 \le r$, $m_1 \le m$.

Let $A=\{j(1), j(2), \dots, j(s')\}$ and $B=\{k(1), k(2), \dots, k(r')\}$ be subsets of $\{1,2,\dots s\}$ and $\{1.2, \dots, r\}$ respectively. Let us consider subsemigroup $^{A}S(K^s)$ in $S(K^s)$ *of* maps sending x_j, $j \in A$ to $f_j(x_{j(1)}, x_{j(2)}, \dots, x_{j(s')})$ and the totality $^{B}S^{s,r}(K)$ in $S^{s,r}(K)$ of maps sending y_j, $j \in B$ to $f_j(y_{k(1)}, y_{k(2)}, \dots, y_{k(r')})$. Totality $^{A}BS_B(K)$ of strings $(H_0, G_1, G_2, H_3, H_4, G_5, G_6, \dots, H_{t-1}, H_t)$ with H_i from $^{A}S(K^s)$ and G_i from $^{B}S^{s,r}(K)$ is a subsemigroup of $^{s}BS_r(K)$. *If* $(H_0, G_1, G_2, H_3, H_4, G_5, G_6, \dots, H_{t-1}, H_t)$ *is in* $^{A}BS_B(K)$ then $(H'_0, G'_1, G'_2, H'_3, H'_4, G'_5, G'_6, \dots, H'_{t-1}, H'_t)$ where G'_i and H'_i are restrictions of G_i and H_i on tuples $(x_{j(1)}, x_{j(2)}, \dots, x_{j(s')})$ and $(y_{k(1)}, y_{k(2)}, \dots, y_{k(r')})$ is in $^{s'}BS_{r'}(K)$. Let symbol μ stand for this homomorphism. We refer to $^{A}BS_B(K)$ as *parabolic semigroup*. The image $^{I}PT(K)$ of $^{A}BS_B(K)$ under the linguistic compression homomorphism ψ of $^{s}BS_r(K)$ onto $^{I}CT(K)$ is a subsemigroup of $^{I}CT(K)$.

Lemma 4. *Let* ψ *and* ψ'' *be the linguistic compression maps of* $^{s}BS_r(K)$ *and* $^{s'}BS_{r'}(K)$ *onto* $^{I}CT(K)$ *and* $^{I'}CT(K)$ *respectively,* ψ' *stands for the restriction of* ψ *onto* $^{A}BS_B(K)$, *arrow between* $^{A}BS_B(K)$ *and* $^{s}BS_r(K)$ *and between* $^{I}PT(K)$ *and* $^{I}CT(K)$ *are natural embedding.* $^{I}PT(K)$ *and* $^{I'}CT(K)$ *are connected by projection homomorphism of point spaces of graphs I and I'. Then the following diagram is commutative.*

$$\begin{array}{ccccc} ^{s'}BS_{r'}(K) & \leftarrow & ^{A}BS_B(K) & \rightarrow & ^{s}BS_r(K) \\ \downarrow \psi'' & & \downarrow \psi' & & \downarrow \psi \\ ^{I'}CT(K) & \leftarrow & ^{I}PT(K) & \rightarrow & ^{I}CT(K) \end{array}$$

4. On semigroups and groups related to Double Schubert graphs and corresponding inverse protocols.

4.1. Construction of graphs, related semigroups and their homomorphisms.

We define Double Schubert Graph $DS(k,K)$ over commutative ring K as incidence structure defined as disjoint union of partition sets $PS=K^{k(k+1)}$ consisting of points which are tuples of kind $x=(x_1, x_2, \dots, x_k, x_{11}, x_{12}, \dots, x_{kk})$ and $LS=K^{k(k+1)}$ consisting of lines which are tuples of kind $y=[y_1, y_2, \dots, y_k, y_{11}, y_{12}, \dots, y_{kk}]$, where x is incident to *y*, if and only if x_{ij} - y_{ij}=x_i y_j for *i=1, 2,..., k* and *j=1, 2,..., k.* It is convenient to assume that the indices of kind *i,j* are placed for tuples of $K^{k(k+1)}$ in the lexicographical order.

Remark.

The term Double Schubert Graph is chosen, because points and lines of $DS(k, F_q)$ can be treated as subspaces of $F_q^{(2k+1)}$ of dimensions *k+1* and *k,* which form two largest Schubert cells. Recall that the largest Schubert cell is the largest orbit of group of unitriangular matrices acting on the variety of subsets of given dimensions. We will consider these connection in details in the next section.

We define the colour of point $x =(x_1 , x_2, \dots , x_k, x_{11} , x_{12}, \dots , x_{kk})$ from *PS* as tuple$(x_1 , x_2, \dots , x_k)$ and the colour of a line $y =[y_1 ,y_2, \dots ,y_k, y_{11} ,y_{12}, \dots ,y_{kk}]$ as the tuple $(y_1 , y_2, \dots ,y_k)$. For each vertex *v* of *DS(k, K),* there is the unique neighbour $y=N_a(v)$ of a given colour $a=(a_1,a_2, \dots ,a_k)$. It means the graphs *DS(k, K)* form a family of linguistic graphs.

Let us consider the subsemigroup $^kY(d, K)$ of $^kBS_k(K)$ consisting of strings $u=(H_0, G_1, G_2, H_3, H_4, G_5, G_6, \dots, H_{t-1}, H_t)$ such that maximum of parameters $deg(H_0)+deg(G_1), deg(G_2)+deg(H_3), deg(H_4)+deg(G_5),$
$deg(G_6)+deg(H_7), deg(G_{t-2})+deg(H_{t-1}), deg(H_t)=1$
equals *d, d>1*.

Theorem 1. *Let I(K) be an incidence relation of Double Schubert graph DS(k, K). Then* $^I\psi(^kY(d, K))={}^kU(d,K)$ *form a family of stable semigroups of degree d.*

The proof is based on the fact that the chain transition *u* from $^kU(d, K)$ moves $x_{i,j}$ into expression $x_{i,j}+T(u)$, where *T(u)* is a linear combination of products $f\epsilon K[x_1, x_2, \dots, x_k]$, $g\epsilon K[y_1, y_2, \dots, y_k]$ where $deg(f)+deg(g)\le d$.
New semigroup $^kU(d, K)$ consists of transformations of the free module K^t, $t=(k+1)k$. If *d=2* then $^kU(d, K)$ contain semigroups of quadratic transformation defined in [9], which consists of ground chain transitions.

Let *J* be subset of the Cartesian square of *M={I,2,…,k}.* We can identify its element *(i,j)* with the index *ij* of Double Schubert Graph *DS(k,K).*

Proposition 1. *Each subset J of* M^2 *defines symplectic homomorphism* δ_J *of DS(k, K) onto linguistic graph* $DS_J(k,K)$.

It is easy to see that in the case of empty set *J* the image of the map is a complete bipartite graph with the vertex set K^kUK^k.

Corollary 1. *Let I(J, K)) be an incidence relation of linguistic graph* $DS_J(k, K)$. *Then* $^{I(J,K)}\psi(^kY(d, K))={}^kU_J(d,K)$ *form a family of stable semigroups of degree d.*

4.2**. Implementation of inverse protocols and their extensions with double Schubert graphs and their symplectic homomorphisms.**

Let us consider the implementation of algorithm 2.1 in the case of *S=S'* and *G=H.* We consider the family of graphs *DS(k, K)* and form the family $DS_{J(k)}(k, K)$. We assume that $j(k)=|J(k)|$ and $c'(k^2)<j(k)<c(k^2)$ for some constants $0<c'<c<1$. We set $S={}^kS={}^I\psi(^kY(d, K))={}^kU(d,K)$ which is a subgroup of affine Cremona group $C(K^n)$, $n=k+k^2$ and $G={}^kG={}^kU_J(d,K)<C(K^m)$, $m=k+j(k)^2$. Alice selects elements $u_i=({}^iH_0, {}^iG_1, {}^iG_2, {}^iH_3, {}^iH_4, {}^iG_5, {}^iG_6, \dots, {}^iH_{t-1}, {}^iH_{t(i)})$, $i=1,2, \dots , r$, $r >1$ of subsemigroup $^kY(d, K)$ and computes $Rev(u_i)$.

She takes $h\epsilon\, {}^kY(d, K)$ together with $Rev(h)$. Alice forms elements u_i and $Rev(u_i)=v_i$ and computes $\varphi(hu_iRev(h))=a'_i$ for $\varphi={}^I\psi$.

She takes f from ${}^kY(d, K)$ and forms strings $fRev(u_i)Rev(f)$. Alice computes ${}^{I(J,K)}\psi(f Rev(u_i)Rev(f))=b'_i$. She takes invertible affine $j=1,2,\dots,t$ transformations T and L of free modules K^n and K^m of kind and forms pairs ($a_i=Ta'_iT^{-1}$, $b_i=L\ b_i\ L^{-1}$) and sends them to Bob.

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

Alice computes u^{-1} as $L\psi(f)f\sigma(\ \varphi(h)^{-1}(T^{-1}wT)\ ^1\varphi(h))\psi(\ f)^{-1})L^{-1}$ where $\psi={}^{I(J,K)}\psi$ and σ homomorphism of ${}^kU(d,K)$ onto ${}^kU_J\ (d,K)$ induced by graph homomorphism δ_J. So Alice and Bob when the protocol ends have mutually inverse encryption/decryption tools u^{-1} and u for the plainspace K^m.

The algorithm is implemented in the cases of $K=Z_p$, $p=2^t$ and $K=F_p$, $p=2^t$ $t=7, 8,\dots, 32$ for $d=2$.

REMARK. Let K be a commutative ring. One can generalise described above algorithm via selection of $h\ \epsilon\ {}^kU(d,Q)$ and f from ${}^kU(d,R)$ where Q and R are extensions of K. Affine transformations T and L have to be chosen among transformations of affine space Q^n and plainspace R^m respectively.

4. 3. Remarks on complexity.

We present complexity estimates in the case of the finite number of used words of length bounded by independent constant and $d=2$. We assume additionally that commutative rings K and R are finite extensions of the ring Q and strings consist of linear polynomials. So Alice can generate symbolic strings in time $O(k)$ .She is able to compute their reverses for $O(k^3)$. Really if $F=(F_1, , F_2, , \dots , F_t)$ is a reversible string then $F_t, {}^{-1}$ can be computed for $O(k^3)$ for the computation of $Rev(F)$. Alice needs t-1 matrix multiplications executed in time $O(k^2)$.

It is easy to see that computation of skating homomorphism requires $O(k^4)$ elementary operations (additions and multiplications) of commutative ring Q. Alice needs to compute images of symplectic projections for several elements. It costs her $O(k^4)$ elementary operations. Additionally she computes composition of linear map and quadratic map of density $O(k^2)$ from k^2+2k variables. Alice can do this in time $O(k^6)$. Finally she has to compute a composition of quadratic and linear map in k^2+2k variables. It takes her $O(k^8)$ operations.

It means that Alice can prepare all data to start algorithm in time $O(k^8)$.

Let us estimate the complexity of computations for Bob. He needs to create two words of finite lengths in corresponding affine Cremona semigroup via several compositions of quadratic polynomials in $n=k^2+2k$ variables. It takes him $O(n^7)$ elementary ring operations. Computation of quadratic map in given point of K^n, $n=k^2+2k$ takes time $O(k^6)$. Thus the total complexity of computations for Bob is $O(n^7)$.

Let us estimate the complexity of decryption process for Alice. She needs computation of product of linear and quadratic maps, product of two quadratic maps of densities $O(k^2)$ and $O(k^4)$**,** product of two quadratic maps of densities $O(k^4)$ and $O(k^2)$**.** It requires $O(k^{10})$ operations.

4.4. On the example with the nonsymplectic homomorphism.

Let $e_1, e_2, \dots, e_k, e_{11}, e_{12}, \dots, e_{kk}$ be natural basis in which graph $DS(k,K)$ is defined. We take an affine space $W(J)$ spanned by $e_1, e_2, \dots, e_l$ and $e_{p,d}$, where $\{p, d\}\epsilon J$ for chosen subset J of Cartesian square of $\{1, 2,\dots, l\}$. We work with the linguistic homomorphism $\phi = \phi_l$ which is simply projection of point (p) an line $[l]$ onto subspace $W(J)$.

Below is the diagram of lemma 4 with $s=r=k$, $A=B=\{1,2,\dots,l\}$, $s'=r'=l$, $I=DS(k, K)$ and $I'=DS_J(l,K)$ is the image of ϕ.

$$\begin{array}{ccccc} {}^{s'}BS_{r'}(K) & \leftarrow & {}^{A}BS_{B}(K) & \rightarrow & {}^{s}BS_{r}(K) \\ \downarrow \psi'' & & \downarrow \psi' & & \downarrow \psi \\ {}^{I'}CT(K) & \leftarrow & {}^{I}PT(K) & \rightarrow & {}^{I}CT(K) \end{array}$$

We define the following implementation of algorithm 2.1, We take the intersection ${}^{kl}Y(d, K)$ of ${}^{k}Y(d, K)$ with ${}^{A}BS_B(K)$. Let α be the homomorphism of ${}^{A}BS_B(K)$ onto ${}^{s'}BS_{r'}(K)$ of the diagram. Alice takes strings $u_i=({}^iH_0, {}^iG_1, {}^iG_2, {}^iH_3, {}^iH_4, {}^iG_5, {}^iG_6, \dots, {}^iH_{t-1}, {}^iH_t)$, $i=1,2, m$ from ${}^{kl}Y(d, K)$ such that first l coordinates of H_t defines bijective map of W. She selects u from ${}^{s}BS_r(K)$ and forms $uu_iRev(u)$. Alice selects v from ${}^{s'}BS_{r'}(K)$ and computes $vRev(\alpha(u_i))Rev(v)$. She computes $a'_i= \psi(uu_iRev(u))$ and $b'_i= \psi''(vRev(\alpha(u_i))Rev(v))$ and sends these pairs to Bob. Alice selects affine transformations a and b *of* affine spaces K^n, $n=k+k^2$ and $K^{l+j(l)}$ respectively. She computes $a_i =aa'_ia^{-1}$ and $b_i =bb'_ib^{-1}$ and sends them to Bob. Alice keeps for herself elements $\psi(u)$, $\psi(Rev(u))$, $\psi''(v)$ and $\psi''(Rev(v))$. The rest of the protocol is going accordingly to general scheme of algorithm 2.1.

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

5. 1. Eulerian groups and and multiplicative linguistic graphs.

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 $x_1 \rightarrow \mu_1 x_1^{a(1,1)}x_2^{a(1,2)} \dots x_n^{a(1,n)}$, $x_2 \rightarrow \mu_2 x_1^{a(2,1)}x_2^{a(2,2)} \dots x_n^{a(2,n)}, \dots, x_n \rightarrow \mu_n x_1^{a(n,1)}x_2^{a(n,2)} \dots x_n^{a(n,n)}$, 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 ${}^nES(K)$. 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}), $g\epsilon G$, and tame Eulerian homomorphisms $\mu: G\rightarrow H$, i. e. computable in polynomial time $t(n)$ homomorphisms of subgroup G of ${}^nEG(K)$ onto $H< {}^mEG(K)$ are motivated by the tasks of Nonlinear Cryptography.

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 ${}^AJG(\pi, \delta)$, where A is triangular matrix with positive integer entries $0\le a(i,j)\le d,\ i\ge d$ defined by the following closed formula.

$y_{\pi(1)}={}_{\mu 1}x_{\delta(1)}{}^{a(1,1)}$

$y_{\pi(2)}={}_{\mu 2}x_{\delta(1)}{}^{a(2,1)}x_{\delta(2)}{}^{a(2,2)}$

…

$y_{\pi(n)}={}_{\mu n}x_{\delta(1)}{}^{a(n,1)}\ x_{\delta(2)}{}^{a(n,2)}\ \ldots x_{\delta(n)}{}^{a(n,n)}$

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

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 refer to the composition of several JG elements as computationally tame multiplicative transformation.

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

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 $x=(x_1, x_2,\ldots, x_s, x_{s+1}, x_{s+2}, \ldots, x_{s+m})Iy=[y_1, y_2, \ldots, y_r, y_{r+1}, y_{r+2}, \ldots, y_{r+s}]$ if and only if $x_2/y_2=g_2w_2(x_1, y_1), x_3/y_3=g_3w_3(x_1, x_2, y_1, y_2), \ldots, x_n/y_n=g_nw_n(x_1, x_2, \ldots, x_{n-1}, y_1, y_2, \ldots, y_{n-1})$, where $g_i \in G$, $i\ge 2$ 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 the tuple of their first coordinates of kind $\boldsymbol{a}=(p_1, p_2, \ldots, p_s)$ or $\boldsymbol{a}=(l_1, l_2, \ldots, l_r)$ 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 jump 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 $J(p,a)=(a_1, a_2, \ldots a_s, p_{1+s}, p_{2+s}, \ldots, p_{s+n})$ and $\rho(J(l,a))=[a_1, a_2, \ldots a_r, p_{1+r}, p_{2+r}, \ldots, p_{r+m}]$. We also consider symplectic and linguistic homomorphisms of linguistic graphs over groups defined similarly to the case of commutative rings.

5.2. Homomorphisms of linguistic compression for semigroups of monomial strings.

Let us use various linguistic graphs over the multiplicative group $G=K^*$ and subsemigroup of monomial strings ${}^{s}BS_r(K^*)$ from ${}^{s}BS_r(K)$, $0<s<n$, $0<r<n$ for generation of pairs of mutually inverse elements of ${}^{n}EG(K)$.

Let us consider the homomorphism of the semigroup ${}^{s}BS_r(K)$ into Eulerian semigroup ${}^{n}ES(K)$, $n=s+m$ defined in terms of linguistic graph $I=I(K^*)$ over K^* of type (s,r,m).

Let N_a be an operator of taking neighbour of given vertex with the colour a in the graph I. Let us consider the commutative group $K'=K^*[x_1, x_2,\ldots,x_s, x_{s+1}, \ldots, x_n]$ of monomial terms of $K[x_1, x_2,\ldots,x_s, x_{s+1}, \ldots, x_n]$ with coefficients from K^* and linguistic graphs I' over group K' defined by the same equations with I but over the larger commutative group K'. We assume that N_a and N'_a are operators of taking neighbour of given vertex with the colour a in the graph I and I' respectively. Let us consider the string of kind $v=(x_1, x_2, \ldots, x_s, x_{s+1}, x_{s+2}, \ldots, x_{s+m})$ from K^{s+m} $(or\ (K')^{s+m})$. We define jump operator ${}^{s}J(v, a)$, $a=(y_1, y_2, \ldots, y_{s,})$ moving v to $(y_1, y_2, \ldots, y_{t,}, x_{s+1}, x_{s+2}, \ldots, x_{s+m})$ from K^{t+m}.

An infinite graph $I'(K')$, $n=m+s$ with partition sets $P'=(K')^{m+s}$ and $L'=(K')^{m+r}$. After that we take a string $u=(H_0, G_1, G_2, H_3, H_4, G_5, G_6, \ldots, H_{t-1}, H_t)$ from ${}^{s}BS_r(K^*)$ and the point $(x)=(x_1, x_2, \ldots, x_n)$ formed by generic elements of K'. This data defines uniquely a skating chain

(x), $J((x),H_0)=({}^1x)$, $N(({}^1x),G_1)=[{}^2x]$, $J([{}^2x],G_2)=[{}^3x]$, $N([{}^3x],H_3)=({}^4x)$, $J(({}^4x),H_4)=({}^5x), \ldots, J([{}^{t-2}x],G_{t-2})=[{}^{t-1}x]$, $N([{}^{t-1}x],H_{t-1})=({}^tx)$, $J(({}^tx),H_t)=({}^tx)$.

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

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

Lemma 5. *The map* $\psi={}^{I}\psi$: ${}^{s}BS_r(K^*) \rightarrow {}^{n}ES(K)$ *is a homomorphism of semigroups.*

We refer to ${}^{I}\psi({}^{s}BS_r(K^*))={}^{I}CT(K^*)$ as a *chain transitions semigroup* of linguistic graph *I(K*) over K** and to map ψ as *multiplicative linguistic compression map*.

5.3. Examples of linguistic graphs over *K**, related semigroups, their homomorphisms and inverse protocol.

We define Double Schubert Graph $DS(k,K^*)$ over the multiplicative group K^* of commutative ring K as incidence structure defined as disjoint union of partition sets $PS^*=(K^*)^{k(k+1)}$ consisting of points which are tuples of kind $x=(x_1, x_2, \ldots, x_k, x_{11}, x_{12}, \ldots, x_{kk})$ and $LS^*=(K^*)^{k(k+1)}$ consisting of lines which are tuples of kind $y=[y_1, y_2, \ldots, y_k, y_{11}, y_{12}, \ldots, y_{kk}]$, where x is incident to y, if and only if $x_{ij} / y_{ij}=x_i y_j$ for $i=1, 2, \ldots, k$ and $j=1, 2, \ldots, k$. It is convenient to assume that the indices of kind i,j are placed for tuples of $(K^*)^{k(k+1)}$ in the lexicographical order.

We define the colour of point $x=(x_1, x_2, \ldots, x_k, x_{11}, x_{12}, \ldots, x_{kk})$ from PS as tuple$(x_1, x_2, \ldots, x_k)$ and the colour of a line $y=[y_1, y_2, \ldots, y_k, y_{11}, y_{12}, \ldots, y_{kk}]$ as the tuple $(y_1, y_2, \ldots, y_k)$. For each vertex v of $DS(k, K^*)$, there is the unique neighbour $y=N_a(v)$ of a given colour $a=(a_1, a_2, \ldots, a_k)$. It means the graphs $DS(k, K^*)$ form a family of linguistic graphs over abelian group K*.

Let $e_1, e_2 \ldots, e_k, e_{11}, e_{12}, \ldots, e_{kk}$ be natural basis in which graph $DS(k,K^*)$ is defined. We take an affine space $W(J)$ spanned by $e_1, e_2 \ldots, e_l$ and $e_{p,d}$, where $\{p, d\} \in J$ for chosen subset J of Cartesian square of $\{1, 2, \ldots, l\}$. We work with the linguistic homomorphism $\phi = \phi_l$ which is simply projection of point (p) an line $[l]$ onto subspace $W(J)$.

Below is the analogue of commutative diagram from section 4 where $A=\{1,2,\ldots,l\}$, $*I=DS(k, K^*)$ and $*I'=DS_J(l, K^*)$ is the image of ϕ, ${}^{A}BS_A(K^*)$ is the totality of strings from ${}^{k}BS_k(K^*)$ with coordinates of kind $(h_1, h_2, \ldots, h_k)$ where $h_i \in K^*[x_1, x_2, \ldots, x_l]$ for $i=1,2,\ldots,l$. Symbol ψ stands for linguistic compression homomorphism of ${}^{k}BS_k(K^*)$ defined by $*I$, ψ' stands for the restriction of ψ onto ${}^{A}BS_A(K^*)$. Map ψ'' is a linguistic compression homomorphism *of* ${}^{l}BS_l(K^*)$ defined by graph $*I'$.

$$
\begin{array}{ccccc}
{}^{l}BS_l(K^*) & \leftarrow & {}^{A}BS_A(K^*) & \rightarrow & {}^{k}BS_k(K^*) \\
\downarrow \psi'' & & \downarrow \psi' & & \downarrow \psi \\
{}^{*I'}CT(K^*) & \leftarrow & {}^{*I}PT(K^*) & \rightarrow & {}^{*I}CT(K^*)
\end{array}
$$

Arrow between ${}^{A}BS_{A}(K^*)$ corresponds to homomorphism μ of projection of each coordinate $(h_1, h_2, \dots, h_k)$ of the string onto the tuple $(h_1, h_2, \dots, h_l)$. Symbol ${}^{*I}PT(K^*)$ stands for $\psi'({}^{A}BS_{A}(K^*))$ **.** Arrow between ${}^{*I}PT(K^*)$ and ${}^{*I'}CT(K^*)$ corresponds to homomorphism of semigroups $*\phi$ induced by linguistic homomorphism ϕ *of* $*I$ *onto* $*I'$.

The protocol. Let ${}^{l}BR_{l}(K^*))$ be a subsemigroup of all reversible strings from ${}^{l}BS_{l}(K^*)$. Alice takes reimage S of μ of ${}^{l}BR_{l}(K^*))$ in the semigroup ${}^{A}BS_{A}(K^*)$. *She* takes strings $u_i=({}^{i}H_0, {}^{i}G_1, {}^{i}G_2, {}^{i}H_3, {}^{i}H_4, {}^{i}G_5, {}^{i}G_6, \dots, {}^{i}H_{t-1}, {}^{i}H_{t(i)})$, $i=1,2,\dots, m$ from S. She selects u from ${}^{k}BR_{k}(K^*)$ and forms $uu_iRev(u)$. Alice selects v from ${}^{l}BR_{l}(K^*))$ and computes $vRev(\mu(u_i))Rev(v)$.She computes $a'_i= \psi(uu_iRev(u))$ and $b'_i= \psi''(vRev(\alpha(u_i))Rev(v))$, selects tame transformations a and b *of from* spaces ${}^{n}EG(K)$, $n=k+k^2$ and ${}^{l+j(l)}EG(K)$ respectively. Alice computes $a_i =aa'_ia^{-1}$ and $b_i =bb'_ib^{-1}$ and sends them to Bob. Alice keeps for herself elements $\psi(u)$, $\psi(Rev(u))$, $\psi''(v)$ and $\psi''(Rev(v))$. The rest of the protocol is going accordingly to general scheme of algorithm 2.2.

6. Conclusion.

The usage of stable inverse platforms was discussed in [4]. For instance correpondents can use cubical collision rules keeping in mind attacks by adversary with the interception of plaintext – ciphertext pairs. In the case of plainspace K^n adversary has to intercept $O(n^3)$ pairs to conduct successful linearization attack in time $O(n^{10})$. Thus correspondents can follow natural recommendation to start a new session of the inverse protocol after the exchange of $O(n^2)$ messages. Instead of a new protocol Alice can use idea of deformation rule. She can use same platform to generate its element g together with its inverse g^{-1}, combine g with two affine bijective maps T_1 and T_2, use her encryption map e_A already elaborated during the session of inverse protocol and send $e_A(T_1gT_2)$ (or $T_1gT_2(e_A)$) to Bob. He can restore T_1gT_2 and use it as the new encryption rule. Alice can decrypt because of her knowledge of the inverse map.

We believe that the case of single toric inverse algorithm has similarity with the case of stable protocol. Adversary has to intercept set of pairs plaintext /ciphertext of polynomial cardinality to interpolate encryption function.

Research on finding of exact upper bounds is an in interesting task. Other interesting question is about the existence of polynomial algorithm to find the inverse of element g from n ${}^{n}EG(K)$ (or ${}^{n}EG'(K)$). Similarly to the problem of finding the inverse of bijective multivariable map a polynomial algorithm to invert g is currently unavailable.

Despite the difference in interpolation of encryption functions security of both toric and stable inverse protocols rests on the same difficult word decomposition problem for the large semigroup, which is intractable with ordinary Turing machine and Quantum Computer.

The usage of tandem which consists of toric and stable inverse protocol allows to create ''eternal'' encryption rule similar to public key but not given publicly. Let us assume that toric and stable protocols of tandem algorithm elaborate pairs of maps $({}^{t}e_A, {}^{s}e_A)$ and $({}^{t}e_B, {}^{s}e_B)$ for Alice and Bob. The problem to interpolate composition ${}^{s}e_A({}^{t}e_A)$, which is non-bijective map of $(K^*)^n$ to K^n of unbounded degree and polynomial density is unfeasible task and decryption function has non polynomial density.

Example of inverse protocols based on toric and stable platforms with outputs acting on $(K^*)^n$ and K^n gives algorithms 5.3 with arbitrary parameter k and l+|J|=n together with algorithm 4.4 with usage graphs $DS(k', K)$ and $DS_{J'}(l',K)$ where $l'+|J'|=n$ and K is a finite field or arithmetic ring. Implementation of different from 4.4 stable algorithms is given in [31], [32], [33], alternative to procedure of 5.3 is given in [6].

Notice that in all mentioned above platforms group enveloped inverse Diffie – Hellman protocol [4] can be used instead of inverse ptotocols 2.1 and 2.2.

References

1. Post-Quantum Cryptography: Call for Proposals:https://csrc.nist.gov/Project; Post-Quantum-Cryptography-Standardization/Call-for-Proposals, Post-Quantum Cryptography: Round 2 Submissions.
2. 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).
3. R. J. McEliece, A Public-Key Cryptosystem Based On Algebraic Coding Theory (1978), DSN Progress Report, 44: 114–116.
4. 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.
5. V. Ustimenko, On the families of stable transformations of large order and their cryptographical applications, Tatra Mt. Math. Publ., 70 (2017), 107-117.
6. V. Ustimenko, On semigroups of multiplicative Cremona transformations and new solutions of Post Quantum Cryptography, Cryptology ePrint Archive, 133, 2019.
7. Max Noerher, Luigi Cremona, Mathematishe Annalen 59, 1904, pp 1-19.
8. 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.
9. V. Ustimenko, On desynchronised multivariate El Gamal algorithm, Cryptology ePrint Archive, 712, 2017.
10. D. N. Moldovyan and 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.
11. L. Sakalauskas, P. Tvarijonas and A. Raulynaitis, Key Agreement Protocol (KAP) Using Conjugacy and Discrete Logarithm Problem in Group Representation Level, INFORMATICA, 2007, vol. 18, No 1, 115-124.
12. 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.
13. 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.
14. Alexei Myasnikov; Vladimir Shpilrain and Alexander Ushakov (2008), Group-based Cryptography, Berlin: Birkhäuser Verlag.
15. Zhenfu Cao (2012), New Directions of Modern Cryptography. Boca Raton: CRC Press, Taylor & Francis Group. ISBN 978-1-4665-0140-9.

16. Benjamin Fine, et. al. "Aspects of Non abelian Group Based Cryptography: A Survey and Open Problems". arXiv:1103.4093.
17. Alexei G. Myasnikov; Vladimir Shpilrain and Alexander Ush akov (2011), Non-commutative Cryptography and Complexity of Group-theoretic Problems, American Mathematical Society.
18. I. Anshel, M. Anshel and D. Goldfeld, An algebraic method for public-key cryptography. Math. Res.Lett. 6(3–4), 287–291 (1999).
19. 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).
20. 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)
21. G. Maze, C. Monico and J. Rosenthal, Public key cryptography based on semigroup actions, Adv.Math. Commun. **1**(4), 489–507 (2007).
22. 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.
23. J.A. Lopez Ramos, J. Rosenthal, D. Schipani and R. Schnyder, Group key management based on semigroup actions, Journal of Algebra and its applications, 2017, vol.16 (08):1750148.
24. 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
25. 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, 15 pages https://doi.org/10.1155/2019/2137561.
26. V. A. Ustimenko, On Schubert cells in Grassmanians and new algorithms of multivariate-cryptography , *Tr. Inst. Mat.*, **23**:2 (2015), 137–148.
27. Vasyl Ustimenko, "On algebraic graph theory and non-bijective multivariate maps in cryptography", *Algebra Discrete Math.*, **20**:1 (2015), 152–170.
28. N. Biggs, Algebraic graphs theory, Second Edition, Cambridge University Press, 1993.
29. V. Ustimenko, Maximality of affine group, hidden graph cryptosystem and graph's stream ciphers, Journal of Algebra and Discrecadete Mathematics, 2005, v.1, pp 51-65.
30. Romańczuk-Polubiec, U. & Ustimenko, V. (2015). On two windows multivariate cryptosystem depending on random parameters // Algebra and Discrete Math., 2015, 19, No. 1, pp. 101-129.
31. 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 998), pp, 654-674.
32. 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.
33. 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.