© Algebra and Discrete Mathematics
Volume **35** (2023). Number 2, pp. 220–250
DOI:10.12958/adm1523

RESEARCH ARTICLE

On new protocols of Noncommutative Cryptography in terms of homomorphism of stable multivariate transformation groups*

V. Ustimenko and M. Klisowski

Communicated by E. I. Zelmanov

Abstract. Noncommutative cryptography is based on applications of algebraic structures like noncommutative groups, semigroups, and noncommutative rings. Its intersection with Multivariate cryptography contains studies of cryptographic applications of subsemigroups and subgroups of affine Cremona semigroups defined over finite commutative rings. Efficiently computed homomorphisms between stable subsemigroups of affine Cremona semigroups can be used in tame homomorphisms protocols schemes and their inverse versions. The implementation scheme with the sequence of subgroups of affine Cremona group that defines the projective limit was already suggested. We present the implementation of another scheme that uses two projective limits which define two different infinite groups and the homomorphism between them. The security of the corresponding algorithm is based on complexity of the decomposition problem for an element of affine Cremona semigroup into a product of given generators. These algorithms may be used in postquantum technologies.

*This research is supported by the fellowship of British Academy for RaR, 2022.

2020 MSC: 11T71, 14G50.

Key words and phrases: multivariate cryptography, stable transformation groups and semigroups, decomposition problem of nonlinear multivariate map into given generators, tame homomorphisms, key exchange protocols, cryptosystems, algebraic graphs.

1. On ideas of Noncommutative Cryptography with platforms of transformations of Multivariate Cryptography

Post Quantum Cryptography serves for the research of asymmetrical cryptographic algorithms which can be potentially resistant against attacks with the usage of a quantum computer. The security of currently popular algorithms is based on the complexity of the following well known three hard problems: integer factorization, discrete logarithm problem, discrete logarithm for elliptic curves. Each of these problems can be solved in polynomial time by Peter Shor's algorithm for the theoretical quantum computer. In fact, some rather old cryptosystems which were suggested in the late '70s of the 20 century potentially may have some resistance to attacks on quantum computers (see for instance McEliece cryptosystem [18]).

Modern PQC is divided into several directions such as Multivariate Cryptography, Nonlinear Cryptography, Lattice-based Cryptography, Hash-based Cryptography, Code-based Cryptography, studies of isogenies for superelliptic curves, Noncommutative cryptography, and others.

The Multivariate Cryptography (see [4, 6, 12]) uses polynomial maps of affine space K^n defined over a finite commutative ring into itself as encryption tools. It exploits the complexity of finding a solution of a system of nonlinear equations from many variables. Multivariate cryptography uses as encryption tools nonlinear polynomial transformations of kind $x_1 \to f_1(x_1, x_2, \ldots, x_n), x_2, \to f_2(x_1, x_2, \ldots, x_n), \ldots, x_n \to f_n(x_1, x_2, \ldots, x_n))$, transforming affine space K^n, where $f_i : K[x_1, x_2, \ldots, x_n]$, $i = 1, 2, \ldots, n$ are multivariate polynomials usually given in a standard form, i.e. via a list of monomials in a chosen order.

Noncommutative cryptography appeared with attempts to apply the Combinatorial group theory to Information Security. If G is a noncommutative group then correspondents can use conjugations of elements involved in the protocol, some algorithms of this kind were suggested in [7, 19, 22, 23], where group G is given with the usage of generators and relations. The security of such algorithms is connected to Conjugacy Search Problem (CSP) and Power Conjugacy Search Problem (PCSP), which combine CSP and Discrete Logarithm Problem and their generalizations. Currently, Noncommutative cryptography is essentially wider than group-based cryptography. It is an active area of cryptology, where the cryptographic primitives and systems are based on algebraic structures like groups, semigroups, and noncommutative rings (see [1–3,5,11,13,17,20,21]).

This direction of security research has very rapid development (see [14,16] and further references in these publications).

One of the earliest applications of noncommutative algebraic structures for cryptographic purposes was the usage of braid groups to develop cryptographic protocols. Later several other noncommutative structures like Tompson 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 cryptographic schemes defined in terms of wide classes of semigroups and their implementations for chosen semigroup families (so-called platform semigroups).

The paper is devoted to some research on the intersection of Noncommutative and Multivariate Cryptographies. We try to use some abstract schemes in terms of Combinatorial Semigroup Theory for the implementation with platforms which are semigroups and groups of polynomial transformations of free modules K^n where K is a commutative ring.

The most popular form of Multivariate cryptosystem is the usage of a single very special map f in a public key mode. The first examples were based on families of quadratic bijective transformation fn(see [4,6,12]), such choice implies a rather fast encryption process. The paper is devoted to other aspects of Multivariate cryptography when some subsemigroup of affine Cremona semigroup of all polynomial transformations is used instead of a single transformation. Let us discuss a case of subsemigroup with a single generator. Everybody knows that Diffie-Hellman key exchange protocol can be formally considered in general case of any finite group or semigroup G. In the case of group G, the corresponding ElGamal cryptosystem can be introduced. Notice that the security of this algorithm depends not only on abstract group G but on the way of its generation in computer memory. For instance, if $G = Z_p^*$ is a multiplicative group of a large prime field then the discrete logarithm problem (DLP) is a difficult one and guarantees the security of the protocol. If the same abstract group is given as an additive group of Z_{p-1} protocol is insecure because DLP will be given by linear equation.

Notice that the implementation of the idea to use a multivariate generator in its standard form has to overcome essential difficulties. At first glance, the Diffie-Hellman protocol in affine Cremona semigroup looks like an unrealistic one because the composition of two maps of degree r and s taken in “general position” will be a transformation of degree rs. So in majority of cases $\deg(F) = d$, $d > 1$ implies very fast growth of function $d(r) = \deg(F^r)$. Of course in the case of the generator in common position, not only a degree but also a density (total number of monomial terms of the map in its standard forms) grows exponentially.

So we have to find special conditions on a subsemigroup of affine Cremona group which guarantees the polynomial complexity of procedure to compute the composition of several elements from subsemigroup. Such conditions can define a basis of Noncommutative Multivariate Cryptography. Hopefully, at least two conditions of this kind are already known [26] (see further references) and [28]. We consider them in the following section.

2. On stable subsemigroups of Affine Cremona Semigroup, Eulerian transformations and corresponding cryptographic scheme

Stability condition demands that the degree of each transformation of the subsemigroup of affine Cremona semigroup has to be bounded by independent constant d. We refer to such subsemigroup as a stable subsemigroup of degree d. Examples of known families of stable subgroups of degree $d = 3$ reader can find in [26] (see further references) or [30]. Applications of such families to Symmetric Cryptography could be found in [32]. Some examples of stable families of subgroups of degree 2 are given in [25].

The eulerian condition demands that all transformations of subsemigroup of affine Cremona subgroup are given in a standard form

$$(x_1, x_2, \ldots, x_n) \\ \to (f_1(x_1, x_2, \ldots, x_n), f_2(x_1, x_2, \ldots, x_n), \ldots, f_n(x_1, x_2, \ldots, x_n))$$

where each f_i has density 1. All transformations of this kind form General Eulerian Semigroup ${}^nGES(K)$ of transformations of kind

$$x_1 \to \mu_1 x_1^{a(1,1)} x_2^{a(1,2)} \ldots x_n^{a(1,n)}, \quad x_2 \to \mu_2 x_1^{a(2,1)} x_2^{a(2,2)} \ldots x_n^{a(2,n)}, \\ \ldots, \quad x_n \to \mu_n x_1^{a(n,1)} x_2^{a(n,2)} \ldots x_n^{a(n,n)}$$

where $a(i, j)$ are positive integers and $\mu_i \in K$.

First cryptosystems of Nonlinear Multivariate Cryptography in terms of ${}^nGES(K)$ are suggested in [28].

The *discrete logarithm problem* is the special simplest case of the *word decomposition problem* for semigroups. Let S' be a subsemigroup of S generated by elements $g_1, g_2, \ldots, g_t$. The *word problem* (WP) of finding the decomposition of $g \in S$ into a product of generators g_i is difficult, i.e. polynomial algorithms to solve it with Turing machine or Quantum Computer are unknown. The idea to apply this problem in Cryptography was considered in [37] where some general schemes to use WP for constructions of algorithms of Noncommutative Cryptography were suggested. Of course, the complexity of the problem depends heavily on the choice of S and the way of a presentation of the semigroup. In the cases of families of affine Cremona semigroups or $S =^n GES(K)$, the problem WP is computationally infeasible with a Turing machine and with Quantum Computer.

We are working on implementations of the following formal schemes of usage of the complexity of WP. Tame map means computable in polynomial time from parameter m.

Toric Tahoma cryptosystem

Let K be a commutative ring, subgroups ${}^nG$ of ${}^nGES(K)$ act naturally on $(K^*)^n$, ${}^mS(n, K)$ is a subsemigroup of ${}^mGES(K)$ such that there is a tame homomorphisn $\Delta = \Delta(m, n)$ of ${}^mS(n, K)$ onto ${}^nG$. We assume that $m = m(n)$ where $m > n$ and consider the following *toric tahoma cryptosystem*:

Alice takes $b_1, b_2, \ldots, b_s$, $s > 1$ from ${}^mS(n, K)$ and $a_1, a_2, \ldots$, as where $a_i = \Delta(b_i)^{-1}$. She takes $g \in^m EG(K)$ and $h \in^n EG(K)$ and forms pairs $(g_i, h_i) = (g^{-1}b_ig, h^{-1}a_ih)$, $i = 1, 2, \ldots, s$ and sends them to Bob.

He writes the word $w(z_1, z_2, \ldots, z_s)$ in the alphabet $z_1, z_2, \ldots, z_s$ together with the reverse word $w'(z_1, z_2, \ldots, z_s)$ formed by characters of w written in the reverse order. He computes element $b = w(g_1, g_2, \ldots, g_s)$ via specialization $z_i = g_i$ and $a = w'(h_1, h_2, \ldots, h_s)$ via specialization $z_i = h_i$. Bob keeps a for himself and sends b to Alice. She computes a^{-1} as $h^{-1}\Delta(gbg^{-1})h$.

Alice writes her message $(p_1, p_2, \ldots, p_n)$ and sends ciphertext $a^{-1}(p_1, p_2, \ldots, p_n)$ to Bob. He decrypts with his function a. Symmetrically Bob sends his ciphertext $a(p_1, p_2, \ldots, p_n)$ to Alice and she decrypts with a^{-1}.

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

The first platforms for this scheme and some other abstract schemes are suggested in [28].

Affine Tahoma cryptosystem

If we change semigroup ${}^{m}GES(K)$ for affine Cremona semigroup $S(K^m)$ we obtain the following *Affine Tahoma Cryptosystem* on stable transformations.

Let K be a commutative ring, stable subgroups n^G of $S(K^n)$ act naturally on K^n and ${}^{m}S(n, K)$ be a subgroup of $S(K^m)$ such that there is a tame homomorphisn $\Delta = \Delta(m, n)$ of ${}^{m}S(n, K)$ onto ${}^{n}G$. We assume that $m = m(n)$ where $m > n$.

Alice takes $b_1, b_2, \ldots, b_s, s > 1$ from ${}^{m}S(n, K)$ and $a_1, a_2, \ldots, a_s$ where $a_i = \Delta(b_i)^{-1}$. She takes $g \in C(Q^m)$ and $h \in C(R^n)$ where R and Q are extensions of the commutative ring K and forms pairs $(g_i, h_i) = (g^{-1}b_i g, h^{-1}a_i h)$, $i = 1, 2, \ldots, s$ and sends them to Bob. We assume that $g = g'T$, $h = h'T'$ where semigroup $\langle g', {}^{m} S(n, K)\rangle$ generated by g' and elements of ${}^{m}S(n, K)$ and group $\langle h', G\rangle$ are stable semigroups of degree d and $T \in AGL_n(R)$, $T' \in AGL_m(Q)$.

As in the previous algorithm Bob writes the word $w(z_1, z_2, \ldots, z_s)$ in the alphabet $z_1, z_2, \ldots, z_s$ together with the reverse word $w'(z_1, z_2, \ldots, z_s)$ formed by characters of w written in the reverse order. He computes element $b = w(g_1, g_2, \ldots, g_s)$ via specialization $z_i = g_i$ and $a = w'(h_1, h_2, \ldots, h_s)$ via specialization $z_i = h_i$. Bob keeps a for himself and sends b to Alice. She computes a^{-1} as $h^{-1}\Delta(gbg^{-1})h$.

Alice writes her message $(p_1, p_2, \ldots, p_n)$ from R^n and sends ciphertext $a^{-1}(p_1, p_2, \ldots, p_n)$ to Bob. He decrypts with his function a. Symmetrically Bob sends his ciphertext $a(p_1, p_2, \ldots, p_n)$ to Alice and she decrypts with a^{-1} (see [27]). Let ${}^{n}TC(K, R, Q)$ stand for affine Tahoma cryptosystem as above.

In [25] quadratic stable subsemigroups with corresponding homomorphisms are suggested as platforms of this scheme. Some other schemes are also implemented there with these platforms. Some cubical platforms were suggested in [27].

Only one family of platforms was investigated via computer implementation. Paper [31] is devoted to implementations of Affine Tahoma scheme with platforms of cubical stable groups. They were defined via families of linguistic graphs that form projective limits and the standard homomorphisms between two members of these sequences. So we have pairs $(G_n, \Delta n)$ where $G_n < S(K^n)$, Δn is a homomorphism of G_n onto G_m,

$m = m(n)$ such that projective limits $\lim(G_n)$, $n \to \infty$ and $\lim(\Delta(G_n))$, $n \to \infty$ coincide with the same infinite transformation group G.

This article is devoted to another computer experiment with the new platform which uses the same groups G_n but different tame homomorphisms η_n . In the new scheme $\lim(G_n)$, $n \to \infty$ equals to G, but $\lim(\eta_n(G_n))$, $n \to \infty$ coincides with the image of homomorphism of G with an infinite kernel.

We believe that the option to vary tame homomorphisms in the chosen sequence of semigroup makes the task of cryptanalytic much more difficult.

We use projective limits $D(K)$ and $A(K)$ of the well known graphs $D(n, K)$ (see [15,33]) and $A(n, K)$ (see [31] and further references) defined over arbitrary finite commutative rings. Walks on the graphs $D(K)$ and $A(K)$ allow to define groups $GD(K)$ and $GA(K)$ of cubic transformations of infinite dimensional affine space over K. Group $GA(K)$ is a homomorphic image of $GD(K)$, both groups can be obtained as projective limits of sequences $GA_n(K)$ and $GD_n(K)$,$n = 1, 2, \dots$ of finite cubical stable groups. We suggest key exchange protocols based on homomorphisms of $GD_j(K)$ onto $GA_i(K)$ for some i and j.

Computer simulations demonstrate an interesting effect of density stabilization of generated cubical maps. The time execution tables for algorithms of generation of maps and numbers of monomial terms are given. They demonstrate the feasibility of algorithms. The method of generation allows constructing for each bijective transformation of the free module over K its inverse map. Multivariate nature of collision maps allows using these algorithms for the safe exchange of multivariate transformations. Various *deformation rules* can be used for this purpose (see formal schemes of [25–27]).

3. Some basic definitions

Let us consider basic algebraic objects of multivariate cryptography, which are important for the choice of appropriate pairs of maps f, f^{-1} in both cases of public key approach or idea of asymmetric algorithms with protected encryption rules. Let us consider the totality $SF_n(K)$ of all rules of kind: $x_1 \to f_1(x_1, x_2, \dots, x_n)$, $x2 \to f_2(x_1, x_2, \dots, x_n)$, $\dots$, $x_n \to f_n(x_1, x_2, \dots, x_n)$ acting on the affine space K^n, where f_i, $i = 1, 2, \dots, n$ are elements of $K[x_1, x_2, \dots, x_n]$ with natural operation of composition. We refer to this semigroup as semigroup of formal transformation $SF_n(K)$ of free module K^n. In fact it is a totality of all endomorphisms of ring $K[x_1, x_2, \dots, x_k]$ with the operation of their superposition. Each rule f from $SF_n(K)$ induces transformation $t(f)$ which sends tuple $(p_1, p_2, \ldots, p_n)$ into $(f_1(p1, p2, \ldots, p_n), f_2(p_1, p_2, \ldots, p_n), \ldots, f_n(p_1, p_2, \ldots, p_n))$. Affine Cremona semigroup $S(K_n)$ is a totality of all transformations of kind $t(f)$. The canonical homomorphism $t \rightarrow t(f)$ maps infinite semigroup $SF_n(K)$ onto finite semigroup $S(K_n)$ in the case of finite commutative ring K.

We refer to pair (f, f') of elements $SF_n(K)$ such that ff' and $f'f$ are two copies of identical rule $x_i \rightarrow x_i$, $i = 1, 2, \ldots, n$ as pair of invertible elements. If (f, f') is such a pair, then product $t(f)t(f')$ is an identity map. Let us consider the subgroup $CF_n(K)$ of all invertible elements of $SF_n(K)$ (group of formal maps). It means f is an element of $CF_n(K)$ if and only if there is f' such that ff' and $f'f$ are identity maps. It is clear that the image of a restriction of t on $CF_n(K)$ is affine Cremona group $C_n(K)$ of all transformations of K^n onto K^n for which there exists a polynomial inverse.

We say that a family of subsemigroups S_n of $SF_n(K)$ (or $S(K_n)$) is stable of degree d if maximal degree of elements from S_n is an independent constant d, $d > 1$. If K is a finite commutative ring then stable semigroup has to be a finite set.

Condition $d > 1$ is natural because of elements from the group $AGL_n(K)$ of all affine bijective transformations, i.e. elements of affine Cremona group of degree 1.

4. On linguistic graphs and related semigroups of affine transformations

The missing definitions of graph-theoretical concepts that appear in this paper can be found in [27]. All graphs we consider are *s*imple 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 vGu 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 xIy 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 $y = [y_1, y_2, \ldots, y_r, y_{r+1}, y_{r+2}, \ldots, y_{r+m}]$ if and only if the following relations hold

$$\begin{aligned} a_1 x_{s+1} + b_1 y_{r+1} &= f_1(x_1, x_2, \ldots, x_s, y_1, y_2, \ldots, y_r) \\ a_2 x_{s+2} + b_2 y_{r+2} &= f_2(x_1, x_2, \ldots, x_s, x_{s+1}, y_1, y_2, \ldots, y_r, y_{r+1}) \\ a_m x_{s+m} + b_m y_{r+m} &= f_m(x_1, x_2, \ldots, x_s, x_{s+1}, \ldots, x_{s+m}, y_1, y_2, \ldots, y_r, \\ &\qquad y_{r+1}, \ldots, y_{r+m}) \end{aligned}$$

where a_j, and b_j, $j = 1, 2, \ldots, m$ are not zero divisors, and f_j are multivariate polynomials with coefficients from K [15]. 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*. Examples of families of linguistic graphs of type $1, 1, m$ and their cryptographic applications can be found in [24], [34] and [36].

We consider also linguistic incidence structures defined by infinite number of equations. Let $M = \{m_1, m_2, \ldots, m_d\}$ be a subset of $\{1, 2, \ldots, m\}$ (set of indexes for equations). Assume that equations indexed by elements from M of following kind

$$\begin{aligned} a_{m_1} x_{m_1} + b_{m_1} y_{m_1} &= f_{m_1}(x_1, x_2, \ldots, x_s, y_1, y_2, \ldots, y_r) \\ a_{m_2} x_{m_2} + b_{m_2} y_{m_2} &= f_{m_2}(x_1, x_2, \ldots, x_s, x_{m_1}, y_1, y_2, \ldots, yr, y_{m_1}) \\ &\cdots \\ a_{m_d} x_{m_d} + b_{m_d} y_{m_d} &= f_{m_d}(x_1, x_2, \ldots, x_s, x_{m_1}, x_{m_2}, \ldots, x_{m_{d-1}}, \\ &\qquad y_1, y_2, \ldots, y_r, y_{m_1}, y_{m_2}, \ldots, ym_{d-1}) \end{aligned}$$

are defined other linguistic incidence structure IM. Then the natural projections $\delta_1 : (x) \to (x_1, x_2, \ldots, x_s, x_{m_1}, x_{m_2}, \ldots, x_{m_d})$ and $\delta_2 : [y] \to [y_1, y_2, \ldots, y_r, y_{m_1}, y_{m_2}, \ldots, y_{m_d}]$ of free modules define the natural homomorphism δ of incidence structure I onto IM. We will use same symbol ρ for the colouring of linguistic graph IM.

It is clear, that δ is a colour-preserving homomorphism of incidence structures (bipartite graphs). We refer to δ as symplectic homomorphism and graph IM as symplectic quotient of linguistic graph I. In the case of linguistic graphs defined by an infinite number of equations, we may consider symplectic quotients defined by infinite subset M (see [33], where symplectic homomorphism was used for the cryptosystem construction).

We consider the more general concept of linguistic homomorphism ξ of linguistic incidence systems $P, L, I(K)$ and induced by linear projections δ of P and δ' of L defined via deleting of some coordinates of colour tuples.

$(x_1, x_1, \ldots, x_s)$ and $[y_1, y_2, \ldots, y_r]$ together with simultaneous deleting of x_{i+r} and y_{i+s} for i from some subset of $\{1, 2, \ldots, m\}$. The image of ξ is a linguistic graph of type s_1, r_1, m_1 where $s_1 \leqslant s$, $r_1 \leqslant r$, $m_1 \leqslant m$.

In the case of linguistic graph Γ the path consisting of its vertices v_0, v_1, v_2, $\ldots$, v_k is uniquely defined by initial vertex v_0, and colours $\rho(v_i,)$, $i = 1, 2, \ldots, k$.

Let us concentrate on linguistic graphs of type $1, 1, m$. Let $N(a, v)$ be the operator of taking neighbour of the vertex v with colour $a \in K$. We refer to sequences $(f_1, f_2, \ldots, f_s)$ with $f_1 \in K[x_1]$ of even length s as symbolic strings. On the totality S1,1 (K) of such sequences we consider the product $(f_1, f_2, \ldots, f_s)(g_1, g_2, \ldots, g_r) = (f_1, f_2, \ldots, f_s, g_1(f_s(x_1)), g_2(f_s(x_1)), \ldots, g_r(f_s(x_1)))$.

Proposition 1. *Elements of $S_{1,1}(K)$ with defined product form a semigroup.*

If Q is an extension of the ground commutative ring K then linguistic graph $I(Q)$ and can be defined via the same set of equations. Let us take $Q = K[x_1, x_2, \ldots, x_n]$ and consider infinite linguistic graph $I' = I(K[x_1, x_2, \ldots, x_n])$ with partition sets P' and L' isomorphic to variety $K[x_1, x_2, \ldots, x_n]^n$. For each symbolic string $(f_1, f_2, \ldots, f_s)$ from $S_{1,1}(K)$ and consider the symbolic computation $C(f_1, f_2, \ldots, f_s)$ which is a walk in I' with starting point $X = (x_1, x_2, \ldots, x_n)$ are generic elements of the commutative ring $K[x_1, x_2, \ldots, x_n]$, other elements of the walk are $X_1 = N(f_1, X)$, $X_2 = N(f_2, X_1)$, $\ldots$, $X_s = N(f_s, X_{s-1})$. Notice that operators $N(f_i, X_{i-1})$ are computed in the graph I'.

It is easy to see that $X_s = (f_s(x_1), g_2(x_1, x_2), \dots, g_n(x_1, x_2, \dots, x_n))$, where $g_i \in K[x_1, x_2, \dots, x_i]$. The rule $(x_1 \to f_s(x_1), x_2 \to g_2(x_1, x_2), \dots, x_n \to g_n(x_1, x_2, \dots, x_n))$ defines the map from $S(K^n)$ into itself. We denote this map as $\Delta I(K)(f_1, f_2, \dots, f_s)$ and refer to it as a map of symbolic computation.

Proposition 2. *A map $\Delta I(K)$ from $S_{1,1}(K)$ into $s(K^n)$ sending symbolic string $(f_1, f_2, \dots, f_s)$ to $\Delta I(K)(f_1, f_2, \dots, f_s)$ is a homomorphism of $S_{1,1}(K)$ into $s(K^n)$.*

We refer to the image $PS(I(K))$ of homomorphism of proposition 2 as semigroup of symbolic point to point computations and refer to $\Delta I(K)$ as linguistic compression (lc) homomorphism. We define a semigroup $LS(I(K))$ of line-to-line computations via simple change of points for lines in I and I'.

Proposition 3. *A symplectic homomorphism δ of linguistic graphs ${}^1I(K)$ and ${}^2I(K)$ of type $(1, 1, n)$ induces canonical homomorphism of $PS({}^1I(K))$ onto $PS({}^2I(K))$.*

Let us consider subsemigroup $\Sigma(K)$ of $S_{1,1}(K)$ generated by symbolic shifting strings of kind $(x_1+a_1, x_1+a_2, \dots, x_1+a_s)$, where $a_i, i = 1, 2, \dots, s$ are elements of K. We identify tuple $C = (x_1 + a_1, x_1 + a_2, \dots, x_1 + a_s)$ with its code $\langle a_1, a_2, \dots, a_s \rangle$.

Proposition 4. *For each linguistic graph $I(K)$ of type $(1, 1, n-1)$ the image $\Sigma(I(K))$ of $\Sigma(K)$ under the linguistic compression homomorphism onto $PS(I(K))$ is a subgroup of affine Cremona group.*

In fact for invertibility of $\delta(f_1, f_2, \dots, f_s) \in PS(I(K))$ the bijectivity of f_s is a sufficient and necessary condition. We refer to $\Sigma(I(K))$ as group of walks on points of linguistic graph $I(K)$.

Let $C = (x_1 + a_1, x_1 + a_2, \dots, x_1 + a_s)$ be a shifting symbolic string from the semigroup $\Sigma(K)$. We refer to $\mathrm{Rev}(C) = (x_1 - a_s + a_{s-1}, x_1 - a_s + a_{s-2}, \dots, x_1 - a_s + a_1, x_1 - a_s)$ as revering string for x.

Lemma. *Let $\Delta = \Delta I(K)$ be linguistic compression map from $S_{1,1}(K)$ onto $PS(I(K))$ and $x \in \Sigma(K)$. Then inverse map for $\Delta(x)$ coincides with $\Delta(\mathrm{Rev}(x))$.*

5. Stable groups of cubical maps defined in terms of linguistic graphs and their homomorphisms

Let K be a commutative ring. We define $A(n, K)$ as bipartite graph with the point set $P = K^n$ and line set $L = K^n$ (two copies of a Cartesian power of K are used). We will use brackets and parenthesis to distinguish tuples from P and L. So $(p) = (p_1, p_2, \ldots, p_n) \in P_n$ and $[l] = [l_1, l_2, \ldots, l_n] \in L_n$. The incidence relation $I = A(n, K)$ (or corresponding bipartite graph I) is given by condition pIl if and only if the equations of the following kind hold

$$p_2 - l_2 = l_1 p_1, \quad p_3 - l_3 = p_1 l_2, \quad p_4 - l_4 = l_1 p_3, \quad p_5 - l_3 = p_1 l_4,$$
$$\ldots \quad , p_n - l_n = p_1 l_{n-1} \text{ for odd } n \text{ and } p_n - l_n = l_1 p_{n-1} \text{ for even } n.$$

Let us consider the case of finite commutative ring K, $|K| = m$. As it instantly follows from the definition the order of our bipartite graph $A(n, K)$ is $2m^n$. The graph is m-regular. In fact the neighbour of given point p is given by above equations, where parameters $p_1, p_2, \ldots, p_n$ are fixed elements of the ring and symbols $l_1, l_2, \ldots, l_n$ are variables. It is easy to see that the value for l_1 could be freely chosen. This choice uniformly establishes values for $l_2, l_3, \ldots, l_n$. So each point has precisely m neighbours. In a similar way, we observe the neighbourhood of the line, which also contains m neighbours. We introduce the colour $\rho(p)$ of the point p and the colour $\rho(l)$ of line l as parameter p_1 and l_1 respectively.

It means that graphs $A(n, K)$ with colouring ρ belong to the class of Γ linguistic graphs of type $(1, 1, n-1)$.

Let $GA(n, K) = \Sigma(A(n, K))$ stands for the group of walks on points of $A(n, K)$. We have a natural homomorphism $GA(n+1, K)$ onto $GA(n, K)$ induced by symplectic homomorphism Δ from $A(n+1, K)$ onto $A(n, K)$ sending point $(x_1, x_2, \ldots, x_n, x_{n+1})$ to $(x_1, x_2, \ldots, x_n)$ and line $[x_1, x_2, \ldots, x_n, x_{n+1}]$ to $[x_1, x_2, \ldots, x_n]$. It means that there is well defined projective limit $A(K)$ of graphs $A(n, K)$ and groups $GA(K)$ of groups $G(n, K)$ when n is growing to infinity. As it stated in [35] case of $K = F_q$, $q > 2$ infinite graph $A(F_q)$ is a tree. Some properties of infinite groups $GA(K)$ of transformation of infinite dimensional affine space over commutative ring K the reader can find in [31].

Other family $D(n, K)$ of linguistic graphs of type $(1, 1, n-1)$ defined over the commutative ring K were defined in [33] but its definition in the case of $K = F_q$ was known earlier. In fact graphs $D(n, q) = D(n, F_q)$ are widely known due to their applications in Extremal Graph Theory, in

Theory of LDPC codes and Cryptography. Graphs $D(n,K)$ are bipartite with set of vertices $V = P \cup L, |P \cap L| = 0$. A subset of the vertices P is called the set of points and another subset L is called the set of lines. Let P and L be two copies of Cartesian power K^n, where $n \geqslant 2$ is an integer. Two types of brackets are used in order to distinguish points from lines. It has a set of vertices (collection of points and lines), which are n-dimensional vectors over $K : (p) = (p_1, p_2, p_3, p_4, \ldots, p_i, p_{i+1}, p_{i+2}, p_{i+3}, \ldots, p_n)$, $[l] = [l_1, l_2, l_3, l_4, \ldots, l_i, l_{i+1}, l_{i+2}, l_{i+3}, \ldots, l_n]$. The point (p) is incident with the line $[l]$, if the following relations between their coordinates hold: $l_2 - p_2 = l_1 p_1$, $l_3 - p_3 = l_2 p_1$, $l_4 - p_4 = l_1 p_2$, $l_i - p_i = l_1 p_{i-2}$, $l_{i+1} - p_{i+1} = l_{i-1} p_1$, $l_{i+2} - p_{i+2} = l_i p_1$, $l_{i+3} - p_{i+3} = l_1 p_{i+1}$ where $i \geqslant 5$. Connected component of edge-transitive graph $D(n,q)$ is denoted by $CD(n,q)$ [15]. Notice that all connected components of the natural projective limit $D(q)$ of graphs $D(n,q)$, $n \to \infty$ are q-regular trees. Let $D(K)$ stands for the projective limit of graphs $D(n,K)$.

Let us denote as $GD(n,K)$ and $GD(K)$ the groups $\Sigma(D(n,K))$ and $\Sigma(D(K))$ of walks on points of graphs $D(n,K)$ and $D(K)$ respectively. For the description of certain symplectic quotients we will use the alternative description of graphs $D(K)$. It 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 $(p) = (p_{0,1}, p_{1,1}, p_{1,2}, p_{2,1}, p_{2,2}, p'_{2,2}, p_{2,3}, \ldots, p_{i,i}, p'_{i,i}, p_{i,i+1}, p_{i+1,i}, \ldots)$, $[l] = [l_{1,0}, l_{1,1}, l_{1,2}, l_{2,1}, l_{2,2}, l'_{2,2}, l_{2,3}, \ldots, l_{i,i}, l'_{i,i}, l_{i,i+1}, l_{i+1,i}, \ldots]$. 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.

$$l_{i,i} - p_{i,i} = l_{1,0} p_{i-1,i}, \quad l'_{i,i} - p'_{i,i} = l_{i,i-1} p_{0,1},$$
$$l_{i,i+1} - p_{i,i+1} = l_{i,i} p_{0,1}, \quad l_{i+1,i} - p_{i+1,i} = l_{1,0} p'_{i,i}.$$

This four relations are defined for $i \geqslant 1$, ($p'_{1,1} = p_{1,1}$, $l'_{1,1} = l_{1,1}$).

Similarly, we can define the projective limit $A(K)$ of graphs $A(n,K)$, $n > 1$.

We can describe the bipartite infinite graph $A(K)$ on the vertex set consisting on points and lines $(p) = (p_{0,1}, p_{1,1}, p_{1,2}, p_{2,1}, p_{2,2}, p_{2,3}, \ldots, p_{i,i}, p_{i,i+1}, \ldots)$. $[l] = [l_{1,0}, l_{1,1}, l_{1,2}, l_{2,1}, l_{2,2}, l_{2,3}, \ldots, l_{i,i}, l_{i,i+1}, \ldots]$ such that point (p) is incident with the line $[l]$ ($(p)I[l]$, if the following relations between their coordinates hold: $l_{i,i} - p_{i,i} = l_{1,0} p_{i-1,i}$, $l_{i,i+1} - p_{i,i+1} = l_{i,i} p_{0,1}$.

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 $D = \{(1,0), (0,1), (1,1), (1,2),$

$(2,2),(2,2)',\ldots,(i-1,i),(i,i-1),(i,i),(i,i)',\ldots)$. So graph $A(K)$ is a symplectic quotient of linguistic incidence structure $D(K)$. Let us use symbol Ψ for the corresponding symplectic homomorphism. For each positive integer $m \geqslant 2$ we consider subsets $M = A^m$ and $M = D^m$ containing of first $m-2$ elements of $A' = A - \{(1,0),(0,1)\}$ and $D' = D - \{(1,0),(0,1)\}$ with respect to the above orders and obtain symplectic quotients I_M of $D(K)$ and $A(K)$. One can check that corresponding quotients are isomorphic to graphs $D(m,K)$ and $A(m,K)$. The investigation of pair A^m, D^m leads to following statement [33].

Proposition 5. *For each $n \geqslant 4$ there are a symplectic homomorphisms of $D(2n,K)$ onto $A(m,k)$, $2 \geqslant m \geqslant n+1$ and $D(2n+1,K)$ onto $A(m,K)$, $2 \geqslant m \geqslant n+2$. Notice that $D(n,K) = A(n,K)$ for $n = 2,3$.*

Proposition 6. *Groups $GD(K)$ and $GA(K)$ are stable cubical transformations of infinite-dimensional affine space over a commutative ring K.*

Corollary. *$GD(n,K)$ and $GA(n,K)$ are stable cubical subgroups of Cremona group $C(K^n)$.*

6. On Three Gates Bridge diagram and algorithms of Noncommutative cryptography for stable transformation groups

Let us consider the following Three Gates Bridge diagram

$$\begin{tikzcd}
\Sigma(R) \arrow[d] & \Sigma(Q) \arrow[l] \arrow[d] & \Sigma(Q) \arrow[l] \arrow[r] \arrow[d] & \Sigma(K) \arrow[d] \\
GA(m,R) & GA(m,Q) \arrow[l] & GD(n,Q) \arrow[l] \arrow[r] & GD(n,K)
\end{tikzcd}$$

Commutative rings K and R are finite extensions of the basic commutative ring Q. Left and rights arrows of the first row of the diagram correspond to the natural embedding of $\Sigma(Q)$ into $\Sigma(R)$ and $\Sigma(K)$. The middle row between two copies of $\Sigma(Q)$ corresponds to identity isomorphism.

Left and rights arrows of the second row of the diagram corresponds to natural embeddings of $GA(m,Q)$ into $GA(m,R)$ and $GD(n,Q)$ into $CD(n,K)$. The middle row between $GD(m,Q)$ and $GA(m,Q)$ corresponds to the homomorphism of these groups induced by symplectic homomorphism of linguistic graphs $D(n,Q)$ and $A(m,Q)$ described in Proposition 5.

Vertical arrows of the diagram correspond to linguistic compression homomorphisms of $S_{1,1}(R)$ onto $PS(I(R))$, $I(R) = A(m, R)$ restricted onto $\Sigma(R)$, $S_{1,1}(Q)$ onto $PS(I(Q))$, $I(Q) = A(m, Q)$ restricted onto $\Sigma(Q)$, $S_{1,1}(Q)$ onto $PS(D(n, Q))$ restricted onto $\Sigma(Q)$, $S_{1,1}(K)$ onto $PS(D(n, K))$ restricted onto $\Sigma(K)$. As it follows from the definitions Three Gates Bridge diagram is commutative diagram.

6.1. Tahoma word protocol

Alice sets pairs of graphs $D(n, Q)$ and its symplectic image $A(m, Q)$. She chooses ring extensions R and K. This information defines Three Bridge Diagram. She selects strings $C_i = \langle {}^i\alpha_1, {}^i\alpha_2, \ldots, {}^i\alpha_{t(1)} \rangle$, $i = 1, 2, \ldots, r$ from $\Sigma(Q)$ and elements $B = \langle \beta_1, \beta_2, \ldots, \beta_s \rangle$ from $\Sigma(K)$ and $D = \langle \gamma_1, \gamma_2, \ldots, \gamma_k \rangle$ from $\Sigma(K)$. Alice computes $\mathrm{Rev}(B)$ and $\mathrm{Rev}(D)$. She takes affine transformations $T_1 \in AGL_n(K)$ and T_2 from $AGL_m(K)$.

Alice forms strings $B_i = \mathrm{Rev}(B)C_iB$ and $D_i = \mathrm{Rev}(D)C_iD$, $i = 1, 2, \ldots, r$ in $\Sigma(K)$ and $\Sigma(R)$. She computes images CB_i and CD_i of linguistic compression homomorphism $\Delta^{D(n,K)}$ and $\Delta^{A(m,K)}$ on elements B_i and D_i. Finally Alice computes elements $T_1^{-1}CB_iT_1 = G_i$ and $F_i = T_2^{-1}CD_iT_2$ which are elements of affine Cremona groups $C(K^n)$ and $C(R^m)$.

Alice keeps the pairs (G_i, F_i) and computes additionally for herself
$H = T_1^{-1}\Delta^{D(n,K)}(\mathrm{Rev}(B))$, $H^{-1} = \Delta^{D(n,K)}(B)T_1$ and
$Z = T_2^{-1}\Delta^{DA(m,K)}(\mathrm{Rev}(D))$, $Z^{-1} = \Delta^{A(m,K)}(D)T_2$.

The homomorphism $\delta : GD(n, Q) \to GA(m, Q)$ of the diagram is tame, i.e. its image can be computed in polynomial time in variable n. The triple $(GD(n, Q), A(m, Q), \delta)$ can be considered as a platform of Tahoma protocol introduced in [27], word tahoma stands for an abbreviation of tame homomorphism.

Tahoma word protocol exchange scheme: Alice uses (G_i, F_i) and pairs (H, H^{-1}) and (Z, Z^{-1}) from affine Cremona groups $C(K^n)$ and $C(R^m)$ as starting data of the following protocol (steps S1–S4)

S1. Alice sends pairs (G_i, F_i), $i = 1, 2, \ldots, r$ to Bob.

S2. Bob takes formal alphabet $A = \{z_1, z_2, \ldots, z_r\}$ and writes a word $w = u_1, u_2, \ldots, u_k$ where $u_i \in A$. He computes the specializations g and f for w of kind $u_j = G_i$ and $u_j = F_i$ if u_j coincides with z_i, $i = 1, 2, \ldots, r$ in groups $\langle G_1, G_2, \ldots, G_r \rangle < GD(n, K) and \langle F_1, F2, \ldots, F_r \rangle < GA(m, R)$ respectively.

S3. Bob sends g to Alice and keeps f for himself.

S4. Alice computes $f_1 = HgH^{-1}$, $f_2 = \delta(f_1)$ and gets collision map f as Zf_2Z^{-1}.

Final remarks Adversary has to find the decomposition of f into generators G_1, G_2,..., G_r. The polynomial algorithms to solve this problem in ordinary Turing machine or Quantum computer are unknown.

Description of the implementation of other protocols of [27] and all complexity estimates are presented in the sections 7 and 8 (see also preprint [29]) together with parameters of computer simulation.

7. Other protocols

7.1. Inverse Tahoma word protocol

Alice changes F_i onto their inverses computed via elements
${}^*D_i = \text{Rev}(D)(\text{Rev}(C_i))D$, ${}^*CD_i = \Delta^{A(m,K)}({}^*D)$ and
${}^*F_i = T_2^{-1}{}^*CD_iT_2$.

Alice sends pairs $(G_i,{}^* F_i)$ to Bob.

As in previous algorithm he takes formal alphabet $A = \{z_1, z_2, \dots, z_r\}$ and writes a word $w = u_1, u_2, \dots, u_k$ where $u_i \in A$. He computes the specializations g or w of kind $u_j = G_i$ and if u_j coincides with z_i, $i = 1, 2, \dots, r$ in groups $\langle G_1, G_2, \dots, G_r\rangle < GD(n, K)$. Bob forms the reverse word ${}^*w = u_k, u_{k-1}, \dots, u_1$. After that he substitutes ${}^*F_i$ and computes corresponding word f in group $\langle F_1, F_2, \dots, F_r\rangle < GA(m, R)$.

Bob send g to Alice. She computes $f_1 = HgH^{-1}$, $f_2 = \delta(f_1)$ and gets map f^{-1} as Zf_2Z^{-1}.

Correspondents can exchange information in secure way. Alice writes message $(p) = (p_1, p_2, \dots, p_m)$, $p_u \in R$ computes cipherext $f^{-1}(p) = (c)$ and sends it to Bob. He decrypts with his map f. In his turn Bob uses f as encryption map and Alice decrypts with her f^{-1}.

7.2. Group enveloped Diffie-Hellman protocol based on homomorphism of $GD(K)$ onto $GA(K)$

Alice uses (G_i, F_i), $i = 1, \dots, r$ and pairs (H, H^{-1}) and (Z, Z^{-1}) from affine Cremona groups $C(K^n)$ and $C(R^m)$ together with ${}^*F_i$. She takes also ${}^*G_i$ computed via elements ${}^*B_i = \text{Rev}(B)(\text{Rev}(C_i))B$, ${}^*CB_i = \Delta D(m, K)(*B)$ and ${}^*G_i = T_1^{-1}{}^*CB_iT_1$. Alice takes string C from $\Sigma(Q)$ and positive integer k_A. She computes symbolic string C^d, $d = k_A$ in $\Sigma(Q)$ and $\Sigma^{D(m,K)}(\text{Rev}(B)CB)$ and $\Delta^{A(m,K)}(\text{Rev}(D)C^dD)$. Finally Alice constructs $G = T_1^{-1}\Delta^{D(m,K)}(\text{Rev}(B)CB)T_1$ and $G_A = T_2^{-1}\Delta^{A(m,K)}(\text{Rev}(D)C^dD)T_2$.

She sends (G_i, F_i), $(^*G_i, ^* F_i)$, $i = 1, \ldots, r$ to Bob together with G and G_A.

Bob selects positive integer $l = k_B$ and word $w = u_1, u_2, \ldots, u_k$. He forms $^*w = u_k, u_{k-1}, \ldots, u_1$.

Bob computes the specializations g or w of kind $u_j = G_i$ and if u_j coincides with z_i, $i = 1, 2, \ldots, r$ in the sub group $\langle G_1, G_2, \ldots, G_r \rangle$ of $GD(n, K)$. He computes g^{-1} as specialization of $^*w$ such that $u_j =^* G_i$ if u_j coincides with z_i, $i = 1, 2, \ldots, r$. Similarly Bob computes the specialization h of w of kind $u = u_j = F_i$ if u_j coincides with z_i and h^{-1} with appropriate specialization of $^*w$.

He computes element $U = g^{-1} G^l g$ and sends it to Alice but keeps for himself $h^{-1} G_A^l h = W$.

Alice can recover the collision map W via computations of $W_1 = HUH^{-1}$, $\delta(W_1) = W_2$, $W_3 = W_2^d$ and $W = ZW_3Z^{-1}$.

Remark 1. Adversary has to find the decomposition of U into generators G, G_1, G_2,..., G_r in the affine Cremona group.

7.3. Inverse group enveloped Diffie-Hellman protocol

This algorithm uses same data.

Alice computes $G_A = T_2^{-1} \Delta^{A(m,K)}(\mathrm{Rev}(D) C^d D) T_2$. but instead of computation of G as $T_1^{-1} \Delta^{D(m,K})(\mathrm{Rev}(B) CB) T_1$ she computes G as $T_1^{-1} \Delta D(m, K)(\mathrm{Rev}(B)(\mathrm{Rev}(C) B) T_1$, i.e., changes G for its inverse. So Bob gets pair (G_A, G) and complete the same steps as in the case of previous algorithm. In this new version he gets the same W but the new element U is an inverse of the map from the previous version.

Alice computes $W_1 = HUH^{-1}$, $\delta(W_1) = W_2$, $W_3 = W_2^d$ and $W_4 = ZW_3Z^{-1}$, but obtained W_4 is the inverse of W.

So in algorithm 7.3 correspondents elaborate mutually inverse maps W (Bob) and $W - 1$ (Alice). Alice writes message $(p) = (p_1, p_2, \ldots, p_m)$, $p_u \in R$ computes cipherext $W^{-1}(p) = (c)$ and sends it to Bob. He decrypts with his map W. In his turn Bob uses W as encryption map and Alice decrypts with her W^{-1}.

So like in the case of 7.1 Alice and Bob can exchange messages in a secure way.

7.4. General complexity estimates for the protocols

Let us assume that Alice is going to use the homomorphism between $D(n, Q)$ and $A(m, Q)$ for $m < n$ and $m = O(n)$. Rings K and R are finite extensions of Q. So we can assume that the cost of arithmetic operation in these commutative rings is $O(1)$. We will count number of arithmetical operations of a commutative ring K which she needs to generate an element of $g = G(n, K)$ which corresponds to symbolic computation with the key of length $O(1)$.

Without loss of generality we may assume that correpondents are involved with Inverse Tahoma Protocol. Counting steps of recurrent process of maps generation via the semigroup compression homomorphisms gives us $O(n)$ operations. Alice chooses already computed affine transformations T and T^{-1}. Alice forms elements b_1, b_2, $\dots$,b_r from $G(n, K)$ together with their inverses and homomorphic images $\mu'(b_i)$, $i = 1, 2, \dots, r$ from $G(m, K)$ in time $O(n)$. She takes T and T^{-1} from $AGL_n(K)$ and forms $a_i = TbiT^{-1}$ and $a'i = T(bi^{-1})T^{-1}$ in time $O(n^7)$.

Bob receives the list of pairs a_i, a'_i, $i = 1, 2, \dots, r$. He computes chosen word of kind $a = a_{i_1}^{k_1} a_{i_2}^{k_2} \dots, ai_t{}^{k_t}$ for chosen finite parameter t and integers k_i, $i = 1, 2, \dots, t$ in time $O(n^{13})$ operations and sends it to Alice. Bob writes his message $p = (p_1, p_2, \dots, p_m)$. To form ciphertext he applies to p transformation a'_{i_t} with multiplicity k_t, a'_{i_t-1} with multiplicity $k_{t-1}, \dots, a'_{i_1}$ with multiplicity k_1 and forms ciphertext c. It takes him $O(n^3)$ elementary operations. Alice computes cubical $b = aT$ with $O(n^5)$ operations. After she gets $d = T^{-1}b$ in time $O(n^7)$. Alice easily gets $\mu(d)$ and computes $e = T_1 d$ and $f = eT_1^{-1}$. She computes p as $f(c)$. The last step cost her $O(n^3)$ elementary ring operations.

Remark 2. The complexity of algorithm execution is $O(n^{13})$. More accurate evaluation in terms of number d of monomial terms in the standard form of cubical maps gives us complexity Cd^4n^{-3}, where C is independent constant.

Studies of parameter d are presented in the next section. Computer simulations demonstrate that in the case of finite fields of characteristic 2 parameter $d = O(n^3)$ and algorithm can be executed in time $O(n^9)$.

Simulations allow us to get similar bound in the cases of arithmetical and Boolean rings.

7.5. On safe exchange of symbolic transformations

The symbolic nature of a collision map can be used for a task that differs from the exchange of keys. We refer to it as the usage of DH *deformation symbolic rules.*

Let Alice have a free module K^n over a commutative ring K. She has a subset Ω of K^n and polynomial map $f : K^n \to K^n$ such the restriction of $f|\Omega$ is an injective map from Ω onto $f(\Omega) = \Gamma$. Additionally, Alice has an algorithm to solve in polynomial time equation $x = b$ with respect to unknown x from Ω and b from Ω.

Alice and Bob use *tahoma word protocol* or symbolic Diffie-Hellman protocol to elaborate the collision map g acting on K^n. After this step Alice sends Ω and transformation $h = f + g$ to Bob. Now Bob can get f as $h - g$. He writes plaintext p from Ω and sends ciphertext $c = f(x)$. Alice uses her data for the decryption.

Remark 3. Notice that the new algorithm is still asymmetrical because Bob can encrypt but not decrypt. The encryption rule is known to the trusted customer (Bob) but an adversary has no access to it. In fact, such access is protected by word problem in semigroup of transformations of K^n or discrete logarithm problem in corresponding affine Cremona semigroup.

Other deformations Alice and Bob agree (via open channel) on a deformation rule $D(f)$ for multivatiate rule f from affine Cremona semigroup. For example, it can be multiplication, i.e. f is the rule $x_i \to f_i(x_1, x_2, \dots, x_n)$,$i = 1, 2, \dots, n$ and g is the rule $x_i \to g_i(x_1, x_2, \dots, x_n)$, $i = 1, 2, \dots, n$ and Alice sends tuple of polynomials f_ig_i, $i = 1, 2, \dots, n$. Bob uses division to restore f. Instead of addition deformation rule (sending of $x_i \to f_i(x_1, x_2, \dots, x_n) + g_i(x_1, x_2, \dots, x_n)$,$i = 1, 2, \dots, n$) Alice can use deformation with adding an element $K[x_1, x_2, \dots, x_n]^n$ obtained from g via the usage of s-time conducted derivation δ^s, where $\delta = d/x_1 + d/dx_2 + \dots + d/dx_n$ (rule $x_i \to f_i(x_1, x_2, \dots, x_n) + \delta^s g_i(x_1, x_2, \dots, x_n)$,$i = 1, 2, \dots, n$). The last deformation is interesting because in many cases we can achieve the equality of degrees for f and $D(f)$. It is easy to continue this list of possible deformation rules.

Remark 4. Let us assume that $\Omega = K^n$. So $f = f(n)$ is a bijection. Assume that degrees of nonlinear maps $f(n)$ are bounded by constant d. Let us assume that the adversary has an option to intercept some plaintext-ciphertext pairs (leakage from Bob's data). In case of interception of $O(n^d)$ adversary has a chance for a successful linearisation attack and get the map f. For example if $d = 3$ then linearisation attack cost is $O(n^{10})$. After that adversary has to find the inverse function for f like in the case of multivariate public key.

To prevent "transition to knowledge" of an encryption multivariate map Alice (or Bob) can arrange a new session with protocol and a transmission of new deformed encryption rule for which secret data for decryption is known.

Remark 5. The technique of linearization attacks on nonbijective maps or maps f_n of unbounded degree and low density is not well developed yet.

8. Graphs $A(n, q)$ and $D(n, q)$, digital condenced matters physics effect

We can substitute graph $A(n, K)$ for other linguistic graph L of type $(1, 1, n-1)$ defined over the commutative ring K and rewrite the content of section 5. We use graphs $A(n, K)$ and well known linguistic graph $D(n, K)$ of this type to implement all algorithm of previous section. Graphs $D(n, K)$ are bipartite with set of vertices $V = P \cup L$, $|P \cap L = 0|$. A subset of the vertices P is called the set of points and another subset L is called the set of lines. Let P and L be two copies of Cartesian power K^n, where $n \geqslant 2$ is an integer. Two types of brackets are used in order to distinguish points from lines. It has a set of vertices (collection of points and lines), which are n-dimensional vectors over K : $(p) = (p_1, p_2, p_3, p_4, \ldots, p_i, p_{i+1}, p_{i+2}, p_{i+3}, \ldots, p_n)$, $[l] = [l_1, l_2, l_3, l_4, \ldots, l_i, l_{i+1}, l_{i+2}, l_{i+3}, \ldots, l_n]$. The point (p) is incident with the line $[l]$, if the following relations between their coordinates hold: $l_2 - p_2 = l_1 p_1$, $l_3 - p_3 = l_2 p_1$, $l_4 - p_4 = l_1 p_2$, $l_i - p_i = l_1 p_{i-2}$, $l_{i+1} - p_{i+1} = l_{i-1} p_1$, $l_i + 2 - p_{i+2} = l_i p_1$, $l_{i+3} - p_{i+3} = l_1 p_{i+1}$ where $i \geqslant 5$. Connected component of edge-transitive graph $D(n, q)$ is denoted by $CD(n, q)$ [15]. Notice that all connected components of the natural projective limit $D(q)$ of graphs $D(n, q)$, $n \to \infty$ infinite graph $D(q)$ are q-regular trees.

Let us denote as $G'(n, K)$ the group of elements of kind $g = \eta(C)$ of *irreducible computation* computation $C = (a_1, a_2, \ldots, a_t)$ in the case of graphs $D(n, K)$.

We present time of generation (in ms) of element g from $G(n, K)$ and $G'(n, K)$ in the cases of graphs $A(n, K)$ and $D(n, K)$ and number $M(g)$ of monomial terms for g.

We refer to parameter t as *length of word.* We can see the "condensed matters physics" digital effect. If t is "sufficiently large", then $M(g)$ is independent from t constant c. It means that the density of cubical collision map in all algorithm is simply c.

We have written a program for generating of elements and for encrypting text using the generated public key. The program is 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 and T_1 are identities,
2) T and T_1 are maps of kind $x_1 \to x_1 + a_2x_2 + a_3x_3 + \ldots + a_tx_t, x_2 \to x_2, x_3 \to x_3, \ldots, x_t \to x_t, a_i \neq 0$, $i = 1, 2, \ldots, t$ (linear time of computing for T and T_1), where $t = n$ and $t = m$, respectively,
3) $T = Ax + b$, $T_1 = A_1x + b_1$; matrices A, A_1 and vectors b, b_1 have mostly nonzero elements.

The tables 8.1–8.6 present the number of monomials depending on the number of variables (n) and the password length in all three cases and both families of graphs $D(n, K)$ and $A(n, K)$, where K is a finite field of characteristic 2. The tables 8.7–8.12 present the time (in milliseconds) of the generation of public key monomials depending on the number of variables n and the length of the word in all three cases and both families of graphs $D(n, K)$ and $A(n, K)$. In [8–10] the similar program for the case when K is Boolean ring was used for investigation of classical Diffie-Hellman protocol for cyclic group $\langle g \rangle$ and corresponding ElGamal cryptosystem. Currently, we expand this computer package on the case of commutative rings Z_m, where m is the power of 2.

Illustrative example Let Alice selects the graph $A(n, K)$ $K = F_{2^{32}}$, $n = 64$ and its canonical homomorphism onto graph $A(32, K)$, which induces canonical homomorphism Δ of $G(64, K)$ onto $G(32, K)$. She takes two irreducible elements of $\Sigma = \Sigma(K)$, $\alpha = (a_1, a_2, \ldots, a_{16})$ and $\beta = (b_1, b_2, \ldots, b_{16})$ of pseudorandom kind, use homomorhism $\eta' = \eta'_{64} of \Sigma$ into $G(64, K)$ and gets elements $a = \eta'(\alpha)$ and $b = \eta'(\beta)$.

Alice forms string $h = (h_1, h_2, \ldots, h_t)$, $t = 16$ and the reverse string $rev(h) = (-h_t + h_{t-1}, -h_t + h_{t-2}, \ldots, -h_t + h_1, -h_t)$ for which $n = \eta'(h) = n$ and $n' = \eta'(rev(h))$.

She takes affine transformation T of the vector space F_q^{64}, $q = 2^{32}$ and its inverse T^{-1} and forms elements $a^1 = Tnan'T^{-1}$ and $b1 = Tnbn'T^{-1}$.

Alice takes $d = (d_1, d_2, \ldots, d_t)$ and the pair $m = \eta'_{32}(d)$, $m' = \eta'_{32}(rev(d))$. She forms $a^2 = Sm\eta'_{32}(\alpha)m'S^{-1} = Sm\eta'_{32}(\alpha)m'S^{-1}$ and $b^2 = Sm\eta'_{32}(\beta)m'S^{-1}$ where S is the bijective affine transformation of 32 dimensional vector space. She sends pairs (a^1, a^2), (b^1, b^2), to Bob. Let us assume that Alice uses transformation T and S of kind 3. It means that cubical transformations a^1 and b^1 are given by lists with 399424 monomial terms and transformations a^2, b^2 are given by their 50720 monomial terms (see table 6).

Bob takes word $w = x^{s_1}y^{r_1}x^{s_2}y^{r_2}\ldots$ of some lengths k, $k \geqslant 3$ (even or odd), where s_1, s_2, …and r_1, r_2, …are positive integers.

He substitutes a^1 and b^1 instead of x and y (or y and x) and compute corresponding transformation c from affine Cremona semigroup of 64 dimensional vector space over finite field F_q. The cubical transformations c is presented by its 388424 monomial terms. Bob substitutes the collision map c' via substitution of a^2 and b^2 in word w instead of x and y. Collision element c' is given by the list of its 50720 monomials.

Bob sends the transformation c to Alice. She computes $c^1 = T^{-1}n'cnT$ which contains 1810 (monomial terms) (see table 4). Alice computes $c^2 = \Delta(c^1)$ given by 770 terms. She reconstructs the collision map as $Smc^2m'S^{-1}$.

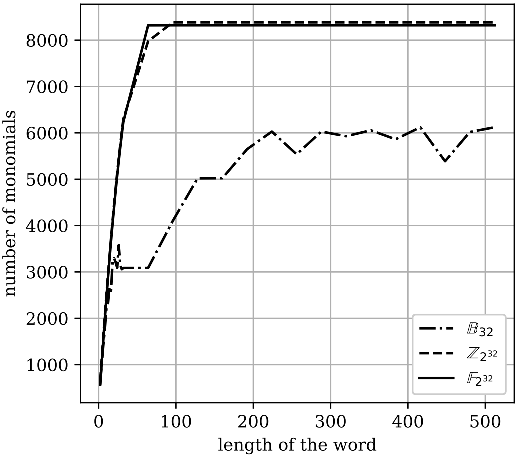

Figure 8.1. Number of monomials in public map ($n = 128$) (graph $D(n, K)$, $K = \mathbb{B}(32), \mathbb{Z}_{2^{32}}, \mathbb{F}_{2^{32}}$), case I

Table 8.1. Number of monomial terms of the cubic map induced by the graph $D(n, \mathbb{F}_{2^{32}})$, case I

	length of the word				
n	16	32	64	128	256
16	145	145	145	145	145
32	544	545	545	545	545
64	1584	2112	2113	2113	2113
128	3664	6240	8320	8321	8321

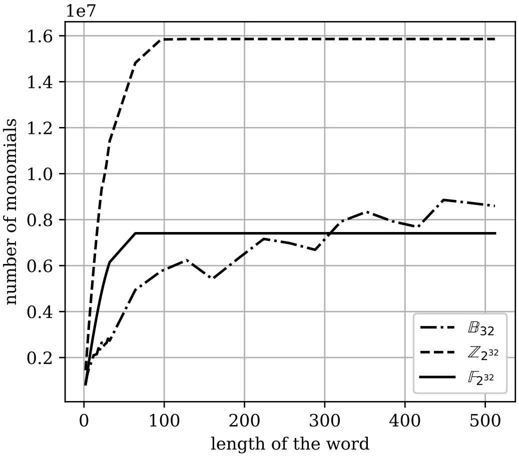

Figure 8.2. Number of monomials in public map ($n = 128$) (graph $D(n, K)$, $K = \mathbb{B}(32), \mathbb{Z}_{2^{32}}, \mathbb{F}_{2^{32}}$), case II

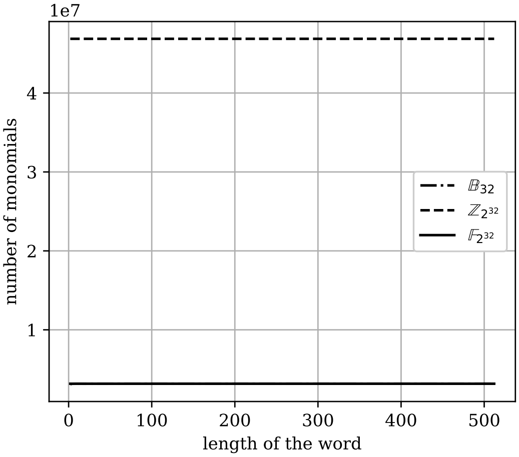

Figure 8.3. Number of monomials in public map ($n = 128$) (graph $D(n, K)$, $K = \mathbb{B}(32), \mathbb{Z}_{2^{32}}, \mathbb{F}_{2^{32}}$), case III

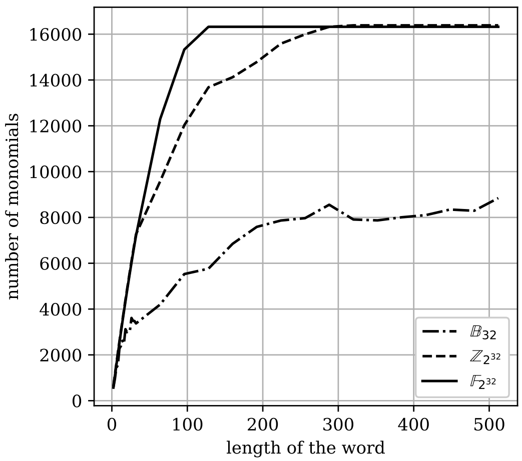

Figure 8.4. Number of monomials in public map ($n = 128$) (graph $A(n, K)$, $K = \mathbb{B}(32), \mathbb{Z}_{2^{32}}, \mathbb{F}_{2^{32}}$), case I

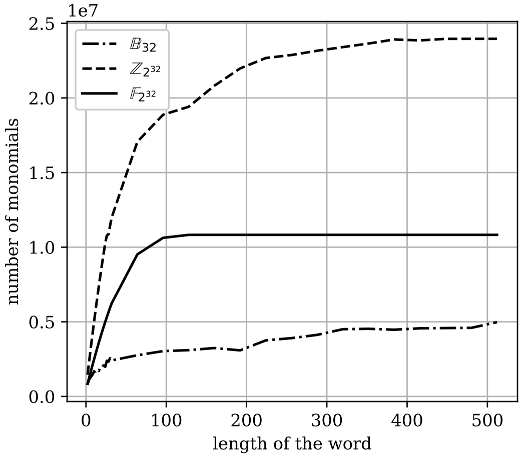

Figure 8.5. Number of monomials in public map ($n = 128$) (graph $A(n, K)$, $K = \mathbb{B}(32), \mathbb{Z}_{2^{32}}, \mathbb{F}_{2^{32}}$), case II

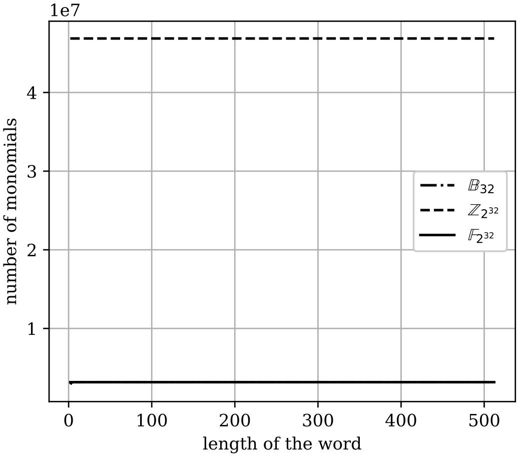

Figure 8.6. Number of monomials in public map ($n = 128$) (graph $A(n, K)$, $K = \mathbb{B}(32), \mathbb{Z}_{2^{32}}, \mathbb{F}_{2^{32}}$), case III

Table 8.2. Number of monomial terms of the cubic map induced by the graph $D(n, \mathbb{F}_{2^{32}})$, case II

	length of the word				
n	16	32	64	128	256
16	3649	3649	3649	3649	3649
32	41355	41356	41356	41356	41356
64	440147	529052	529053	529053	529053
128	3823600	6149213	7405944	7405945	7405945

Table 8.3. Number of monomial terms of the cubic map induced by the graph $D(n, \mathbb{F}_{2^{32}})$, case III

	length of the word				
n	16	32	64	128	256
16	6544	6544	6544	6544	6544
32	50720	50720	50720	50720	50720
64	399424	399424	399424	399424	399424
128	3170432	3170432	3170432	3170432	3170432

Table 8.4. Number of monomial terms of the cubic map induced by the graph $A(n, \mathbb{F}_{2^{32}})$, case I

	length of the word				
n	16	32	64	128	256
16	250	250	250	250	250
32	770	1010	1010	1010	1010
64	1810	3074	4066	4066	4066
128	3890	7202	12290	16322	16322

Table 8.5. Number of monomial terms of the cubic map induced by the graph $A(n, \mathbb{F}_{2^{32}})$, case II

	length of the word				
n	16	32	64	128	256
16	5623	5623	5623	5623	5623
32	53581	62252	62252	62252	62252
64	454375	680750	781087	781087	781087
128	3607741	6237144	9519921	10826616	10826616

Table 8.6. Number of monomial terms of the cubic map induced by the graph $A(n, \mathbb{F}_{2^{32}})$, case III

	length of the word				
n	16	32	64	128	256
16	6544	6544	6544	6544	6544
32	50720	50720	50720	50720	50720
64	399424	399424	399424	399424	399424
128	3170432	3170432	3170432	3170432	3170432

Table 8.7. Public map generation time (ms), $D(n, \mathbb{F}_{2^{32}})$, case I

	length of the word				
n	16	32	64	128	256
16	12	24	32	52	100
32	64	140	292	592	1192
64	1044	2261	4833	9985	20270
128	15821	33846	74340	160213	331895

Table 8.8. Public map generation time (ms), $D(n, \mathbb{F}_{2^{32}})$, case II

	length of the word				
n	16	32	64	128	256
16	28	48	100	212	420
32	284	648	1372	2816	5712
64	3229	8397	19454	41568	85783
128	55075	139366	357361	824166	1758059

Table 8.9. Public map generation time (ms), $D(n, \mathbb{F}_{2^{32}})$, case III

	length of the word				
n	16	32	64	128	256
16	76	140	268	524	1036
32	1224	2328	4541	8968	17828
64	21889	40417	77480	151592	299844
128	453798	812140	1526713	2946022	5792889

Table 8.10. Public map generation time (ms), $A(n, \mathbb{F}_{2^{32}})$, case I

	length of the word				
n	16	32	64	128	256
16	4	12	24	48	96
32	56	132	288	600	1232
64	996	2100	4644	10068	20933
128	15645	33489	74244	167454	364707

Table 8.11. Public map generation time (ms), $A(n, \mathbb{F}_{2^{32}})$, case II

	length of the word				
n	16	32	64	128	256
16	20	60	128	260	540
32	308	788	1776	3760	7716
64	3193	8858	23231	53196	113148
128	54031	137201	368460	950849	2164037

Table 8.12. Public map generation time (ms), $A(n, \mathbb{F}_{2^{32}})$, case III

	length of the word				
n	16	32	64	128	256
16	76	148	288	576	1148
32	1268	2420	4700	9268	18405
64	22144	40948	78551	153784	304240
128	460200	819498	1532277	2970743	5836938

Conclusion

We propose Post Quantum Cryptography information security solutions based on the complexity of the following problem Cremona Semigroup Word Decomposition (CSWD).

Thus we hope that introduced algorithms can be considered as serious candidates to be postquantum cryptographical tools. We believe that future studies of cryptanalytics confirm that CSWD problem remains unsolvable on ordinary Turing Machine and Quantum Computer under the condition of stability of platform S. Hope that the idea of an alternative disclosure of hidden homomorphism will attract the attention of cryptanalytics.

Complexity estimates for both correspondents demonstrate the possibility of the current usage of algorithms. Computer simulations demonstrate an interesting phase-transition effect that allows predicting the density of the collision maps of key exchange protocols and their inverse forms. This effect also demonstrates the feasibility of proposed cryptographic schemes. Direct and inverse protocols to elaborate collision multivariate transformation of free module K^n of predictable density can be used together with stream cipher working with data written in alphabet K or passwords written in this alphabet.

Correspondents can use collision maps to add them to part of a password or part of a plaintext or part of a ciphertext. There is an option to deform part of passwords, plaintext and ciphertext by outcomes of inverse protocols.

References

[1] M. Anshel, M. Anshel, and D. Goldfeld. An algebraic method for public-key cryptography. *Math. Res. Lett.*, 6:287–291, 1999.

[2] S. Blackburn and S. Galbraith. Cryptanalysis of two cryptosystems based on group actions. In K. Lam, C. Xing, and E. Okamoto, editors, *Advances in Cryptology – ASIACRYPT '99*, Lecture Notes in Computer Science, pages 52–61. Springer, 1999.

[3] Z. Cao. *New Directions of Modern Cryptography.* CRC Press, 2012.

[4] J. Ding, J. E. Gower, and D. S. Schmidt. *Multivariate Public Key Cryptosystems.* Advances in Information Security. Springer, 2006.

[5] B. Fine, M. Habeeb, D. Kahrobaei, and G. Rosenberger. Aspects of nonabelian group based cryptography: A survey and open problems. arXiv:1103.4093 [cs.CR], 2011. http://arxiv.org/.

[6] L. Goubin, J. Patarin, and B.-Y. Yang. Multivariate cryptography. In *Encyclopedia of Cryptography and Security*, pages 824–828. Springer US, Boston, MA, 2011.

[7] D. Kahrobaei and B. Khan. A non-commutative generalization of elgamal key exchange using polycyclic groups. In *IEEE GLOBECOM 2006 - 2006 Global Telecommunications Conference*, 12 2006.

[8] M. Klisowski. *Zwiększenie bezpieczeństwa kryptograficznych algorytmów wielu zmiennych bazujących na algebraicznej teorii grafów.* PhD thesis, Politechnika Częstochowska, 2015.

[9] M. Klisowski and V. Ustimenko. On the comparison of cryptographical properties of two different families of graphs with large cycle indicator. *Mathematics in Computer Science*, 6(2):181–198, 2012.

[10] M. Klisowski and V. Ustimenko. Graph based cubical multivariate maps and their cryptographical applications. In L. Beshaj, T. Shaska, and E. Zhupa, editors, *Advances on Superelliptic Curves and their Applications*, volume 41 of *NATO Science for Peace and Security Series - D: Information and Communication Security*, pages 305–327. IOS Press, 2015.

[11] 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 M. Bellare, editor, *Advances in Cryptology — CRYPTO 2000*, pages 166–183, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[12] N. Koblitz. *Algebraic Aspects of Cryptography.* Springer-Verlag, Berlin, Heidelberg, 1998.

[13] P. H. Kropholler, S. J. Pride, W. A. M. Othman, K. B. Wong, and P. C. Wong. Properties of certain semigroups and their potential as platforms for cryptosystems. *Semigroup Forum*, 81(1):172–186, 2010.

[14] G. Kumar and H. Saini. Novel noncommutative cryptography scheme using extra special group. *Security and Communication Networks*, 2017:1–21, 01 2017.

[15] F. Lazebnik, V. A. Ustimenko, and A. J. Woldar. A new series of dense graphs of high girth. *Bull. Amer. Math. Soc.*, 32:73–79, 1995.

[16] J. A. Lopez-Ramos, J. Rosenthal, D. Schipani, and R. Schnyder. Group key management based on semigroup actions. *J. Algebra Appl.*, 16(8), 2017.

[17] G. Maze, C. Monico, and J. Rosenthal. Public key cryptography based on semigroup actions. *Adv. Math. Commun.*, 1(4):489–507, 2007.

[18] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. *DSN Progress Report*, 44:114–116, Jan 1978.

[19] D. N. Moldovyan and N. A. Moldovyan. A new hard problem over non-commutative finite groups for cryptographic protocols. In I. Kotenko and V. Skormin, editors, *Computer Network Security*, pages 183–194, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[20] A. Myasnikov, V. Shpilrain, and A. Ushakov. *Group-based Cryptography.* Advanced Courses in Mathematics — CRM Barcelona. Springer Basel AG, 2008.

[21] A. Myasnikov, V. Shpilrain, and A. Ushakov. *Non-commutative Cryptography and Complexity of Group-theoretic Problems.* Mathematical surveys and monographs. American Mathematical Society, 2011.

[22] E. Sakalauskas, P. Tvarijonas, and A. Raulynaitis. Key agreement protocol (kap) using conjugacy and discrete logarithm problems in group representation level. *Informatica, Lith. Acad. Sci.*, 18:115–124, 01 2007.

[23] V. Shpilrain and A. Ushakov. The conjugacy search problem in public key cryptography: Unnecessary and insufficient. *Applicable Algebra in Engineering, Communication and Computing*, 17(3):285–289, 2006.

[24] V. Ustimenko. On linguistic dynamical systems, families of graphs of large girth, and cryptography. *J. Math. Sci.*, 140(3):461–471, 2007.

[25] V. Ustimenko. On desynchronised multivariate el gamal algorithm. Cryptology ePrint Archive, Report 2017/712, 2017. https://eprint.iacr.org/2017/712.

[26] V. Ustimenko. On the families of stable multivariate transformations of large order and their cryptographical applications. *Tatra Mt. Math Publ.*, 70:107–117, 2017.

[27] V. Ustimenko. On new symbolic key exchange protocols and cryptosystems based on a hidden tame homomorphism. *Reports of the National Academy of Sciences of Ukraine*, (10):26–36, 2018.

[28] V. Ustimenko. On semigroups of multiplicative cremona transformations and new solutions of post quantum cryptography. Cryptology ePrint Archive, Report 2019/133, 2019. https://eprint.iacr.org/2019/133.

[29] V. Ustimenko and M. Klisowski. On noncommutative cryptography and homomorphism of stable cubical multivariate transformation groups of infinite dimensional affine spaces. Cryptology ePrint Archive, Report 2019/593, 2019. https://eprint.iacr.org/2019/593.

[30] V. Ustimenko and M. Klisowski. On noncommutative cryptography with cubical multivariate maps of predictable density. In K. Arai, R. Bhatia, and S. Kapoor, editors, *Intelligent Computing: Proceedings of the 2019 Computing Conference, Volume 2*, number 998 in Advances in Intelligent Systems and Computing, pages 654–674. Springer, 2019.

[31] V. Ustimenko and U. Romańczuk. On extremal graph theory, explicit algebraic constructions of extremal graphs and corresponding turing encryption machines. In *Artificial Intelligence, Evolutionary Computing and Metaheuristics*, pages 257–285. Springer, 2013.

[32] V. Ustimenko, U. Romańczuk-Polubiec, A. Wróblewska, M. K. Polak, and E. Zhupa. On the constructions of new symmetric ciphers based on nonbijective multivariate maps of prescribed degree. *Secur. Commun. Netw.*, 2019, 2019.

[33] V. A. Ustimenko. Coordinatization of regular tree and its quotients. In P. Engel and H. Syta, editors, *Voronoï's Impact on Modern Science*, number 2 in Proceedings of the institute of mathematics of the national academy of sciences of Ukraine. Institute of Mathematics, National Academy of Sciences of Ukraine, 1998.

[34] V. A. Ustimenko. Graphs with special arcs and cryptography. *Acta Applicandae Mathematicae*, 74, 2002.

[35] V. A. Ustimenko. Maximality of affine group, and hidden graph cryptosystems. *Alg. Dis. Mthm.*, 2005(1):133–150, 2005.

[36] U. V. A. On graph-based cryptography and symbolic computations. *Serdica Journal of Computing*, 1(2):131–156, 2007.

[37] N. R. Wagner and M. R. Magyarik. A public key cryptosystem based on the word problem. In *Proceedings of CRYPTO 84 on Advances in Cryptology*, pages 19–36, New York, NY, USA, 1985. Springer-Verlag New York, Inc.

Contact information

Vasyl Ustimenko

Royal Holloway University of London, Egham Hill, Egham TW20 0EX, UK
E-Mail(s): Vasyl.Ustymenko@rhul.ac.uk
Web-page(s): www.royalholloway.ac.uk

Michał Klisowski

University of Maria Curie-Skłodowska,
5 M. Curie-Skłodowskiej Square,
20-031 Lublin, Poland
E-Mail(s): michal.klisowski@mail.umcs.pl
Web-page(s): www.umcs.pl

Received by the editors: 08.01.2020
and in final form 22.06.2023.