New Cryptosystems of Noncommutative Cryptography based on Eulerian Semigroups of Multivariate Transformations

Vasyl Ustimenko[a,b] and Oleksandr Pustovit[b]

[a] *University of Marie Curie-Sklodowska in Lublin, 5 Plac Marii Curie-Skłodowskiej str., Lublin, 20031, Poland*
[b] *Institute of Telecommunications and the Global Information Space of the National Academy of Sciences of Ukraine, 13 Chokolivsky ave., Kyiv, 02000, Ukraine*

Abstract

The intersection of Commutative and Multivariate cryptography contains studies of cryptographic applications of subsemigroups an90d subgroups of affine Cremona semigroups defined over finite commutative ring K with the unity. We consider special subsemigroups in a semigroup of all endomorphisms of $K[x_1, x_2, \ldots, x_n]$ which preserve the variety $(K^*)^n$. Efficiently computed homomorphisms between such subsemigroups can be used in Post Quantum key exchange protocols and their inverse versions when correspondents elaborate mutually inverse transformations of $(K^*)^n$. The security of these schemes is based on the complexity of the decomposition problem for an element of a semigroup into a product of given generators. We suggest two public key cryptosystems for which security rests on the complexity of the problem to compute the inverse of a given semigroup element. The usage of the protocols allows converting these encryption schemes into new cryptosystems which are not public keys and have some similarity with El Gamal cryptosystem. New protocols can be used for the Post Quantum key exchange for one-time pad encryption.

Keywords

Noncommutative cryptography, multivariate cryptography, key exchange protocol, inverse protocols, semigroup of transformations, decomposition problem, public-key cryptosystem.

1. Post Quantum, Multivariate, and Noncommutative Cryptography

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 standardization process [1]. In March 2019, NIST published a list of candidates qualified for the second round of the standardization process. Few public key candidates are implemented, like candidate called Round 5 [2] or the classic McEliece algorithm [3]. In July 2020 the third round was started, only rainbow-like oil and vinegar digital signatures are among selected algorithms of multivariate cryptography. They use nonbijective quadratic maps of affine space and can not be used as encryption algorithms. The outcome of the NIST competition stimulates innovative research on studies of cryptographical applications of multivariate maps of unbounded degree and studies of multivariate cryptosystems which are not public keys.

In this publication, we continue to develop new cryptosystems within an alternative approach [4–6] 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 problems for groups was considered in [8].

Cybersecurity Providing in Information and Telecommunication Systems, January 28, 2021, Kyiv, Ukraine
EMAIL: vasulustimenko@yahoo.pl (A.1); sanyk_set@ukr.net (B.2)
ORCID: 0000-0002-2138-2357 (A.1); 0000-0002-3232-1787 (B.2)

In the inverse protocols considered in this paper, the encryption rule is not given publicly. We introduce new cryptosystems defined in terms of semigroups of transformations of affine space K^n which consist of transformations moving variable to a single monomial term.

One of the directions of PQC is multivariate cryptography [9] which 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 to a system of nonlinear equations from many variables. Multivariate cryptography uses as encryption tools nonlinear polynomial transformations of kind $x_1 \to f_1(x_1, x_2,\dots,x_n)$, $x_2 \to f_2(x_1, x_2,\dots,x_n)$, …, $x_n \to f_n(x_1, x_2,\dots,x_n)$ transforming affine space K^n, where $f_i \in K[x_1, x_2,\dots,x_n]$, $i=1,2,\dots,n$ are multivariate polynomials usually given in the standard form, i. e. via a list of monomials in a chosen order.

We are going to present new crypto algorithms in the area of the intersection of multivariate cryptography and non-commutative cryptography which appeared with attempts to apply 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 [10–13], where group G is given with the usage of generators and relations. 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.

The extension of group-based cryptography is essentially wider direction of 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 [14–18, 20, 21]. This direction of security research has very rapid development [22, 23] and further references in these publications). 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). This direction must be well supported by cryptanalytic research [29–33] 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).

Papers [4], [5], and [6] contain some modifications of the Diffie-Hellman protocol when G is given as a subgroup of affine Cremona semigroup $S(K^n)$ over finite commutative ring K of all polynomial transformations. These papers use the assumption that each element is given in its standard form of multivariate cryptography. To use semigroup operation one has to compute the composition of transformations. This was an attempt to combine methods of non-commutative cryptography and multivariate cryptography.

Paper [4] suggests some usage of homomorphisms of subsemigroups of affine Cremona groups for protocols and cryptosystems which are not generalizations of the Diffie-Hellman algorithm and its El Gamal type modifications. Some examples are given there. The implementations of these schemes with an evaluation of densities of involved polynomial transformations are described in [6]. Elements of graph-based stable subgroups used in [6] can serve as encryption tools of stream ciphers (see [25] and further references).

The current paper aims to apply formal schemes of [4] to the case of transformations of variety $(K^*)^n$, where K^* is a multiplicative group of commutative ring K with unity such that $|K^*| > 1$. Natural examples of such rings are $K=Z_m$ and $K=F_q$ where $m>2$, $q>2$.

We present the new post-quantum key exchange protocols and cryptosystems of El Gamal type of non-commutative cryptography which uses homomorphisms of two semigroups acting on $(K^*)^n$ (3.1-3.4), some straightforward algorithms without the usage of homomorphisms are given in [26]. We hope that some of the presented algorithms will be used in the post-quantum future.

2. On Eulerian Semigroup and Hard Computational Problem

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

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

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

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

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 a triangular matrix with positive integer entries $0 \leq a(i,j) \leq d$, $i \geq d$ defined by the following closed formula.

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

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

We refer to ${}^AJG(\pi, \delta)$ as Jordan - Gauss multiplicative transformation or simply JG element. It is an invertible element of ${}^nES(K)$ with the inverse of kind ${}^BJG(\delta, \pi)$ such that $a(i,i)b(i,i)=1 \pmod d$. Notice that in the case $K=Z_m$ straightforward process of computation the inverse of JG element is connected with the factorization problem of integer m. If $n=1$ and m are a product of two large primes p and q the complexity of the problem is used in RSA public key algorithm. The idea to use the composition of JG elements or their generalizations with injective maps $of\ K^n$ into K^n was used in [27] ($K=Z_m$) and [28] ($K=F_q$).

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

We refer to elements of ${}^nES(K)$ as multiplicative Cremona elements. Assume that the order of K is a constant. As it follows from the definition the computation of the value of element from ${}^nES(K)$ on the given element of K^n is estimated by $O(n^2)$. The product of two multiplicative Cremona elements can be computed in time $O(n^4)$.

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

Remark 2.1. Let G be a subgroup of ${}^nEG(K)$ generated by Jordan-Gauss elements $g_1, g_2, \dots, g_t$. The *word problem* of finding the decomposition of $g \in G$ into the product of generator g_i is difficult, i. e. polynomial algorithms to solve it with Turing machine or Quantum Computer is unknown. If the word problem is solved and the inverses of g_i are computable then the inverse of g is determined. Notice that if $n=1$, $K=Z_m$, $m=pq$ where p and q are large primes and G is generated by $g_1=\mu g_1^a$ the problem is unsolvable with the Turing machine but it can be solved with Quantum Computer.

Each element of the semigroup ${}^nES(K)$ is written in the chosen basis $e_1, e_2,\dots, e_n$. Let $J=\{i(1), i(2),\dots, i(k)\}$ be a subset of $\{1,2,..,n\}$ and $W_J = \langle e_{i(1)}, e_{i(2)},\dots, e_{i(k)}\rangle$ be a corresponding symplectic subspace . We refer to totality ${}^nP_J(K)$ of maps $F \in {}^nES(K)$ preserving W_J as parabolic semigroup of ${}^nES(K)$. The map F from ${}^nP_J(K)$ transforms tuple $(x_{i(1)}, x_{i(2)}, \dots, x_{i(n)})$ accordingly to the rule $x_{i(1)} \to \mu_{i(1)} x_{i(1)}^{a(1,1)} x_{i(2)}^{a(1,2)} \dots x_{i(k)}^{a(1,k)}$, $x_{i(2)} \to \mu_{i(2)} x_{i(1)}^{a(2,1)} x_{i(2)}^{a(2,2)} \dots x_{i(k)}^{a(2,k)}, \dots, x_{i(k)} \to \mu_{i(k)} x_{i(1)}^{a(k,1)} x_{i(2)}^{a(k,2)} \dots x_{i(k)}^{a(2,k)}$.

Let π_J be the restriction of element F from ${}^nP_J(K)$ onto W_J. The map π_J defines the canonical homomorphism of ${}^nP_J(K)$ onto ${}^kES(K)$. If Q is an extension of K we can consider semigroup ${}^nP_{J,K}(Q)$ of maps from ${}^nES(Q)$ transforming $(x_{i(1)}, x_{i(2)}, \ldots, x_{i(n)})$ accordingly to written above rule. The restriction of map $F \epsilon {}^nP_{J,K}(Q)$ on W_J defines homomorphism $\pi_{J,K}$ from ${}^nP_{J,K}(Q)$ onto ${}^kES(K)$.

3. Protocols and Cryptosystems in Terms of Semigroup nES(K)

Let us consider some protocols and cryptosystems based on the idea of a hidden canonical homomorphism. Notice that if commutative ring K' is an extension of K then embedding of K into K' defines canonical embedding of ${}^nES(K)$ into ${}^nES(K')$. Let ${}^nJG(K)$ stand for the totality of all Jordan-Gauss transformations from ${}^nES(K)$.

3.1 Tahoma Protocol

Alice takes finite extensions Q and R of $K \epsilon \{Z_{m.}, F_q\}$ and J of cardinality k and consider a *zigzag diagram*

$$\begin{array}{ccc} {}^nP_{J,K}(Q) & \rightarrow & {}^nES(Q) \\ \downarrow & & \\ {}^kES(R) & \leftarrow & {}^kES(K) \end{array}$$

The horizontal arrows correspond to embeddings of semigroups, vertical arrow corresponds to $\pi_{J,K.}$. Alice takes elements $h_1, h_2, \ldots, h_s$ from ${}^kES(K)$ and creates elements $ext(h_i)$ from their $\pi_{J,K.}$reimages via adding the rules $x_j \rightarrow \mu_j x_1^{a(j,1)} x_2^{a(j,2)} \ldots x_n^{a(j,n)}$ where $\mu_j \epsilon Q^*$ and j is not an element of J. She selects set $S= \{g_1, g_2, \ldots, g_t\}$ of Jordan-Gauss elements g_i, $i=1,2, \ldots,t$ in ${}^nES(Q)$ and word in alphabet S to form tame element w of subgroup $G = <S>$ of ${}^nES(Q)$ together with w^{-1}. Similarly Alice takes Jordan-Gauss generators $S'=\{u_1, u_2, \ldots, u_r\}$ in ${}^kES(R)$, selects word in alphabet S' and forms tame element $u \epsilon <S'>$ and its inverse u^{-1}. She forms pairs $(a_i = w^{-1}ext(h_i)w, b_i = u^{-1}(h_i)u)$, $i=1, 2,\ldots,s$ and sends them to Bob. He takes formal alphabet $Z=\{z_1, z_2, \ldots, z_s\}$ and writes word $w_B=v(z_1, z_2, \ldots, z_s)$ in Z of length d, $d>s$ and computes specialization $z_i=a_i$ and $z_i=b_i$ and takes resulting elements $a=v(a_1, a_2, \ldots, a_s) \epsilon {}^nES(Q)$ and $b=v(b_1, b_2, \ldots, b_s) \epsilon {}^kES(R)$ respectively. Bob keeps b for himself and sends a to Alice.

Alice computes ${}^1a=wa\,w^{-1}$. She takes ${}^2a=\pi_{J,K.}({}^1a)$ and obtains collision element b as $u^{-1}({}^2a)u$.

3.2. Inverse Tahoma Protocol

As in the previous protocol, Alice works with the presented above zigzag diagram**.** She selects sets of Jordan - Gauss generators S in ${}^nES(Q)$ and S' in ${}^kES(R)$ to construct pairs of tame elements w, w^{-1} and u, u^{-1}. Now she takes set ${}^1S$ of Jordan Gauss elements over R from ${}^kES(K) \cap {}^kJG(R)$ and forms elements $h_1, h_2, \ldots, h_s$ from $<{}^1S>$ and their inverses $h_1^{-1} h_2^{-1}, \ldots, h_s^{-1}$ in ${}^kEG(R)$. Notice that elements h_i^{-1}, $i=1,2,\ldots,s$ are elements of ${}^kES(K)$ and larger semigroups ${}^kES(R)$ and ${}^kES(Q)$.

Alice forms $ext(h_i)$ in ${}^nES(Q)$. In the new algorithm she computes pairs $(a_i = w^{-1}ext(h_i)w, b_i = u^{-1}(h_i^{-1})u)$, $i=1, 2,\ldots,s$ and sends them to Bob. He takes formal alphabet $Z=\{z_1, z_2, \ldots, z_s\}$ and writes word $w_B=v(z_1, z_2, \ldots, z_s)=(u_1, u_2, \ldots, u_d)$ in Z of length d, $d>s$ together with the *reverse* word $Rev(w_B)=(u_d, u_{d-1}, \ldots, u_1)$. Bob computes the specialization $z_i=a_i$ of word w_B and $z_i=b_i$ of word $Rev(w_B)$ and takes resulting elements $a=v(a_1, a_2, \ldots, a_s) \epsilon {}^nES(Q)$ and $b=v(b_1, b_2, \ldots, b_s) \epsilon {}^kES(R)$ respectively. Notice that $b \epsilon {}^kEG(R)$. He sends a to Alice and keeps b for himself. Alice computes ${}^1a=wa\,w^{-1}$. She takes ${}^2a=\pi_{J,K.}({}^1a)$ and obtains element b^{-1} as $u^{-1}({}^2a)u$.

Remark. Alice and Bob can securely communicate in the following way. Alice writes a message as a string of characters $(p_1, p_2, \ldots, p_k)$ in alphabet R^* encrypts it by application of b^{-1}. Bob decrypts it with his transformation b.

Similarly, Bob uses b for the encryption of his message from the plain space $(R^*)^k$ and Alice decrypts it with b^{-1}.

3.3. Group Enveloped Diffie-Hellman Key Exchange Protocol

As in the inverse protocol of the previous unit Alice works with the presented above zigzag diagram**.** She selects sets. For simplicity assume that $Q=K=R$. Alice selects sets of Jordan - Gauss

generators S in ${}^{n}ES(K)$ and S' in ${}^{k}ES(K)$ to construct pairs of tame elements w, w^{-1} and u, u^{-1}. Now she takes set ${}^{1}S$ of Jordan Gauss elements over K from ${}^{k}ES(K)$ and forms elements $h_1, h_2, \ldots, h_s$ from $<{}^{1}S>$ and their inverses $h_1^{-1}\ h_2^{-1}, \ldots, h_s^{-1}$ in ${}^{k}EG(K)$. Alice takes $g \in {}^{k}ES(K)$ and positive integer parameter k_A. Alice creates elements $ext(h_i)$, $ext(h_i^{-1})$ and $ext(g)$ from their π_J reimages via adding the rules $x_j \rightarrow \mu_j x_1^{a(j,1)} x_2^{a(j,2)} \ldots x_n^{a(j,n)}$ where $\mu_j \in K^*$ and j is not an element of J. She forms pairs $(a_i = w^{-1} ext(h_i) w,\ b_i = u^{-1}(h_i)\ u),\ i=1, 2, \ldots, s$ and sends them to Bob together with pairs (a_i^{-1}, b_i^{-1}), $g_A = u^{-1} g^l\ u$, $l = k_A$ and $g' = w^{-1} ext(g) w$.

Bob takes formal alphabet $Z=\{z_1, z_2, \ldots, z_s\}$ and writes word $w_B = v(z_1, z_2, \ldots, z_s) = (u_1, u_2, \ldots, u_d)$ in Z of length d, $d>s$ together with the *reverse* word $Rev(w_B) = (u_d, u_{d-1}, \ldots, u_1)$. Bob computes the specialization $z_i = a_i$ of word w_B and $z_i = a_i^{-1}$ of $Rev(w_B)$ and writes resulting elements a and a^{-1} from ${}^{n}ES(K)$. Similarly, he creates b and b^{-1} via specialization $z_i = b_i$ of w_B and specialization and $z_i = b_i^{-1}$ of word $Rev(w_B)$ in the group ${}^{k}EG(K)$ respectively. Bob takes his natural integer k_B. He computes ${}^{B}g = a^{-1} g^d a$, $d = k_B$ and sends it to Alice, and keeps the collision map $c = b^{-1} g_A^{\ d} b$, $d = k_B$. Alice computes *the* collision map as $u^{-1}(\pi_J (w\ {}^{B}g\ w^{-1}))^{l}\ u$, $l = k_A$.

Remark 3.1. The adversary has to decompose ${}^{B}g$ into a_i and g'. After that he/she has to substitute g_A instead of g' and b_i instead of a_i.

3.4. The Inverse Version of Group Enveloped Diffie-Hellman Key Exchange Protocol

Assume that $K=R=Q$ and Alice works with the simplified zigzag diagram ${}^{k}ES(R) = {}^{k}ES(K)$. She forms the same data as in the case of 3.4 but $g \in {}^{k}ES(K)$ has to be invertible. So Alice takes *an* additional set ${}^{2}S$ of Jordan-Gauss elements from ${}^{k}EG(K)$ and forms pair of kind (g, g^{-1}), $g \in <{}^{2}S>$. She sends Bob pairs (a_i^{-1}, b_i^{-1}), $g_A = u^{-1} g^l\ u$, $l = k_A$ and $g^* = w^{-1} ext(g^{-1}) w$ instead of g' of 3.4.

Bob uses a word in the alphabet of formal variables and generates elements a and a^{-1} from ${}^{n}EG(K)$ and b, $b^{-1} \in {}^{k}EG(K)$ in the same way with the case of 3.4 and takes his natural integer k_B. Now he computes ${}^{B}g = a^{-1} g^{*d} a$, $d = k_B$ and sends it to Alice and keeps the map $f = b^{-1} g_A^{\ d} b$, $d = k_B$. Alice computes the inverse map for f as $u^{-1}(\pi_J (w\ {}^{B}g\ w^{-1}))^{l}\ u$, $l = k_A$.

Remark 3.2. Alice and Bob have bijective transformations f and f^{-1} of the variety $(K^*)^{k}$. So they can exchange messages written in the alphabet.

4. New Cryptosystems related to the Semigroup of Eulerian Transformations and Their Transition into Private Mode

Algorithm 4.1. Let us assume that K is a finite commutative group with unity with nontrivial group K^*. Several Jordan - Gauss transformations $J_1, J_2, \ldots, J_s$ from $EG_n(K)$ of kind

$$x_1 \rightarrow b_1 x_1^{a(1,1)}$$

$$x_2 \rightarrow b_2 x_1^{a(2,1)} x_2^{a(2,2)}$$

$$\ldots$$

$$x_n \rightarrow b_n x_1^{a(n,1)} x_2^{a(n,2)} \ldots x_n^{a(n,n)},$$

where $a(i, j) \in Z_m$, $m=|K^*|$, $(a(i,i), m)=1$ are selected by Alice. She can select $s+1$ monomial bijective affine transformations ${}^{1}T, {}^{2}T, \ldots, {}^{s+1}T$ from $AGL_n(K)$ and use

$$G = {}^{1}T\ J_1\ {}^{2}T\ J_2\ {}^{3}T\ J_3 \ldots {}^{s}T\ J_s\ {}^{s+1}T. \qquad (2)$$

Noteworthy that the knowledge of decomposition (2) allows her to find the inverse of G via straightforward algorithms of computing inverses of J_i and ${}^{j}T$. Alice computes G in its standard form and sends it to Bob. Correspondents use $(K^*)^n$ *as* plain space and cipher space.

Bob writes plain space $(p)=(p_1, p_2, \ldots, p_n)$ computes the ciphertext $G((p))$, and sends it to Alice. She decrypts because of her knowledge of decomposition (1).

Algorithm 4.2. Correspondents use plain space $M_n(Z_m)$ of square matrices $(a(i,j))$, $i=1,2,\ldots,n$, $j=1,2,\ldots,n$ of size n over arithmetic ring Z_m of residues modulo $m=|K^*|$. Cipherspace is formed by elements of ${}^{n}ES(K)$.

Alice chooses invertible transformations B and C of kind (1) formed via selected decompositions of kind 2. She sends standard forms of B and C to Bob.

Bob writes his message in the alphabet Z_m in the form of matrix A=*(a(i,j))* and pseudo-random parameters μ_i, $i=1,2,\ldots, n$ to write transformation G written as (1). Bob computes the composition $D=BAC$ and sends it to Alice. She computes $B^{-1}DC^{-1}$ and gets plaintext *A.*

Remark 4.1. We can generalize the algorithm via the selection of μ_i, $i=1,2,\ldots, n$ from $K-\{0\}$.

5. Symbiotic Combination of Protocols with One-Time Pads

Noteworthy that cryptosystem 3.1 and 3.3 can be combined with a one-time pad similar to the classical combination of Diffie-Hellman key exchange protocol and absolutely secure encryption in the sense of C. Shannon.

Let us consider the case $K=Z_m$, $m=2^r$. Group Z^*_m is formed by all odd residues $r(j)=2j+1$, $j=0, 1, \ldots, 2(2^{r-2})+1$.

Let j be a residue of $Z_{m'}$, $m'=2^{r-1}$. The map $\pi(r(j))=j$ is a bijection of $(Z^*)_m$ onto $Z_{m'}$.

So correspondents use protocol 3.1 or 3.3. So they are elaborate collision element G of ${}^nES(K)$ in the form (I). The change μ_i for $\pi(\mu_i)=a_i$ allows us to identify G with the password tuple (a)= $(a_1, a(1,1), a(1,2),\ldots, a(1,n), a_2, a(2,1), \ldots, a(2,n),\ldots,a_n, a(n,1), a(n,2),\ldots, a(n,n))$ from $(Z_{m'})^{n(n+1)}=V$. So correspondents can use V as plainspace, Bob writes plaintext $((p_1, p(1,1), p(1,2),\ldots, p(1,n), p_2, p(2,1), \ldots, p(2,n),\ldots,p_n, p(n,1), p(n,2),\ldots, p(n,n))$ and add password tuple (a) to form ciphertext $((p_1 + a_1, p(1,1)+ a(1,1), p(1,2)+a(2,2),\ldots, p(1,n)+a(1,n), p_2+a_1, p(2,1)+a(2,1), \ldots, p(2,n)+a(2,n),\ldots,p_n + a_n, p(n,1)+a(n,1), p(n,2)+a(n,2),\ldots, p(n,n)+a(n,n))$.

Thus we can use $m=2^8$, $m'=2^7$ and encrypt files with extension .txt. or use $m=2^9$, $m'=2^8$ to encrypt various files in the binary alphabet. We use pairs $(2^{33}, 2^{32})$ and $(2^{65}, 2^{64})$ to work with numerical data in popular alphabets of residues modulo 2^{32} or 2^{64}.

We can use prime field F_{257} to work with binary files. F^*_{257} is a cyclic group of order 256. So correspondents use ${}^nES(F_{257})$. Let us consider bijection π between F^*_{257} and Z_{256} given by rule $\pi(x \bmod 257) =x-1 \bmod 256$. Then we can identify G with the element of $(Z_{256})^t$, $t=n(n+1)$ and use described above combination of the protocols and one-time pad based on the group $(Z_{256}, +)$. Similarly, they can use next Fermat's prime number $2^{16}+1=65537$. They conduct the protocol in ${}^nES(F_{65537})$ and work with a one-time pad with alphabet Z_q, $q=2^{16}$. Correspondents could not use the next two Fermat numbers $2^{32}+1$ and $2^{64}+1$ because they are composite integers (the smallest one has prime factors 641 and 6708417).

Correspondents can work with ${}^nES(F_3)$, use $\pi(x \bmod 3)=x-1 \bmod 2$, and encrypt tuples over F_2. Alternatively, they can use protocols defined for ${}^nES(F_4)$ to work with classical one-time pads over F_2. Another option is the usage of finite fields of characteristic two and works with ${}^nES(F_q)$, $q=2^m$. In this case order of F_q is Mersenne number 2^m-1. We can identify element of F^*_q of kind $a_{m-1}x^{m-1} + a_{m-2}x^{m-2} +..+ a_1x + a_0$ with $(a_{m-1}2^{m-1} + a_{m-2}2^{m-2} +..+ a_12 +a_0)-1$ of Z_d, $d=2^m -1$ and work with one time pad over the additive group of this ring. These one time pads are especially attractive in the case of Mersenne primes $2^m -1$ (for $m=2,3,7,13, 17,19, 31, 61, 89,107, 127,\ldots)$.

Remark 5.1. We can easily switch from one time pad over $(Z_q,+)$ and $(F_q,+)$, $q=2^r$ which isomorphic to additive subgroup of vector space $(F_2)^r$. The natural bijection π is the map sending $a_{r-1}x^{r-1}+a_{r-2}x^{r-2}+ +a_1x+\ldots+ a_0$ from F_q to $a_{r-1}2^{r-1}+a_{r-2}2^{r-2}+ +a_12+\ldots+ a_0 \pmod{2^r}$.

Remark 5.2. Correspondents can use inverse protocols 3.2 and 3.4. to get keys for one-time pad encryption. If Alice gets H and Bob receives H^{r-1}. Then Alice can generate "pseudorandom G" and sends HG (or HGH) to Bob and he restores G.

6. Conversion of Algorithm 4.1 to Cryptosystem of El Gamal Type

Option 1. Correspondents use one of the protocols 3.1 and 3.3 based on the semigroup ${}^nES(K)$, and described above homomorphism of ${}^{m(n)}ES(K)$, where $m(n)$ some linear expression in variable n such that $m(n)>n$. Recall that the security of these protocols rests on the complexity of word problems in the semigroup.

So they elaborate the collision element $H \in {}^nES(K)$ written in the form $x_i \to h_i(x_1, x_2,\ldots,x_n)$, $i=1,2,\ldots,n$ where h_i are monomial terms from $K[x_1, x_2,\ldots, x_n]$.

Alice creates map G of Algorithm 4.1 via decomposition of kind (2) written as

$x_i \rightarrow g_i(x_1, x_2, \ldots, x_n)$. She sends the tuple $(h_1g_1, h_2g_2, \ldots, h_ng_n)$ with all components written in standard form to Bob. He restores g_i, $i=1,2,\ldots,n$ and correspondents start algorithm 4.1.

Option 2. Correspondents use homomorphism of ${}^{m(n)}ES(K)$ onto ${}^{m}EG(K)$. They use protocol 4.2. and elaborate mutually inverse transformations G^{-1} (in the possession of Alice) and G (on the Bob side). So Bob uses G for the encryption of the plaintext (p) and Alice uses her G^{-1} to decrypt.

Option 3. Alice selects expression $G'={}^{1}T\,J_{\underline{1}}\,{}^{2}T\,J_{\underline{2}}\,{}^{3}T\,J_{\underline{3}}\ldots{}^{s}T\,J_{\underline{s}}\,{}^{s+1}T$. She sends $F= G^{-1}$ G' (or G^{-1} G'G^{-1}). Bob computes G' as GF (or GFG). He uses G' as an encryption tool. Alice uses her knowledge on the decomposition of kind (2) and decrypts.

Remark 6.1. Adversaries can use some cyber-terrorist tools to get more than n pairs of kind (plaintext (p_i), corresponding ciphertext (c_i)). It allows him to start the investigation of equations $G(p_i)=c_i$, $i=1,2,\ldots,t$, $t>n$ for the approximation of G (or G'). Noteworthy that the polynomial algorithm for solving this approximation problem is unknown.

Remark 6.2. Noteworthy that correspondents can restrict the number of exchanged messages by $[n/2]$. This restriction does not allow an adversary to collect sufficient data for the approximation process. When the number of Bob's messages rich $[n/2]$ correspondents they start a new protocol of kind 4.1 to establish a new encryption rule for Bob.

Noteworthy that in the presented schemes correspondents can use protocol 3.3 instead of 3.1 and protocol 3.4 instead of 3.2.

7. Conversion of Algorithm 4.2 to a Cryptosystem of El Gamal Type

Option 1. Correspondents use twice data for protocols of kind 3.1 or 3.3 based on the semigroup ${}^{n}ES(K)$, and described above homomorphism of ${}^{m(n)}ES(K)$, where $m(n)$ some linear expression in variable n such that $m(n)>n$.

So correspondents elaborate two collision elements H and H' from ${}^{n}ES(K)$ written in the form $x_i \rightarrow h_i(x_1, x_2, \ldots, x_i)$ and $x_i \rightarrow h'_i(x_1, x_2, \ldots, x_i)$ $i=1,2,\ldots,n$ where h_i, h'_i are monomial terms from $K[x_1, x_2, \ldots, x_n]$. Alice uses different sequences of Jordan-Gauss elements and different sequences of linear monomial transformations to construct two bijective transformations B and C from ${}^{n}EG(K)$ in their standard forms $x_i \rightarrow b_i(x_1, x_2, \ldots, x_i)$ and $x_i \rightarrow c_i(x_1, x_2, \ldots, x_i)$ together with their decompositions of kind (2). So she computes B^{-1} and C^{-1} presented in their standard forms together with corresponding decompositions. Alice sends tuples $(h_1b_1, h_2b_2, \ldots, h_nb_n)$ and $(h'_1c_1, h'_2c'_2, \ldots, h'_nb'_n)$ to Bob. He restores transformations B and C and uses Algorithm 4.2 with these data.

All suggested algorithms were implemented in the case of finite fields $K=F_q$ and arithmetic rings Z_q, $q=2^8, 2^{16}, 2^{32}$.

Remark 7.1. Adversary has to intercept many pairs of kind (plaintext p_i, corresponding ciphertext c_i), $i=1,2,\ldots,t$. He/she can consider transformations

$$Z: x_1 \rightarrow z_1 x_1^{\,z(1,1)} x_2^{\,z(1,2)} \ldots x_n^{\,z(1,n)},\ x_2 \rightarrow z_2 x_1^{\,z(2,1)} x_2^{\,z(2,2)} \ldots x_n^{\,z(2,n)},$$
$$\ldots,\ x_n \rightarrow z_n x_1^{\,z(n,1)} x_2^{\,z(n,2)} \ldots x_n^{\,z(n,n)} \text{ and } Y: x_1 \rightarrow y_1 x_1^{\,y(1,1)} x_2^{\,y(1,2)} \ldots x_n^{\,y(1,n)}$$
$$, x_2 \rightarrow y_2 x_1^{\,y(2,1)} x_2^{\,y(2,2)} \ldots x_n^{\,y(2,n)}, \ldots, x_n \rightarrow y_n x_1^{\,y(n,1)} x_2^{\,y(n,2)} \ldots x_n^{\,y(n,n)}.$$

where z_i, y_i, $z(i,j)$, $y(i,j)$ are variables. Adversary writes intercepted ciphertects $C(l)$ as transformation of kind (1) with $\mu_i = {}^{l}c_i$, $i=1,2,\ldots,n$ and $a(i,j)={}^{l}c(i,j)$. He/she forms elements $P(l)$ of ${}^{n}ES(K)$ from intercepted plaintexts p_l with unknown $\mu_i = {}^{l}u_i$, $i=1,2,\ldots,n$ and $a(i,j)={}^{l}p(i,j)$. Adversary has to investigate nonlinear system of equations $XP(l)Y=C(l)$, $l=1,2,\ldots,t$.

Each equation in the group ${}^{n}ES(K)$ corresponds to n^2equalities in Z_d and n equalities for variables from K^*.

We have $2n^2$ variables in Z_d and $2n+tn$ variables which are parameters from K^*. Simple counting of numbers of variables from Z_d demonstrates that interception of one message does not allow an adversary to compute transformations B and C. So Alice and Bob can exchange two messages safely and start a new session of the protocol.

So we have a postquantum version of the combination Diffie Hellman key exchange protocol and an absolutely secure one-time pad. The advantage is that we can use the encryption scheme at least two times.

In the case of arbitrary t the number of unknown parameters from K^* is larger than a number of equations in terms of ring elements. It means that the usage of pseudorandom parameters in the algorithm allows avoiding new session of the protocol.

8. Conclusion

We present two public-key cryptosystems (Algorithms 4.1 and 4.2) for which security rests on the complexity of finding the inverse for an element from ${}^{n}EG(K)$. The polynomial algorithm for the solution to this problem is unknown. So we hope that the suggested PK system is an interesting one for cryptanalytics.

The combinations of Algorithm 4.1, 4.2, and one-time pads with postquantum protocols of Section 3 produce cryptosystems that are not public keys. They have a certain similarity with the number-theoretic El Gamal cryptosystem.

Security of these algorithms as well as security of corresponding protocols rest on the known hard problem of word decomposition. Noteworthy that used platforms ${}^{n}EG(K)$ are highly noncommutative semigroup for which word decomposition problem is *NP*-hard. So these two cryptosystems can be used as post-quantum instruments.

9. References

[1] Post-Quantum Cryptography: Call for Proposals.URL: https://csrc.nist.gov/Project; Post-Quantum-Cryptography-Standardization/Call-for-Proposals.

[2] M. Andrzejczak, The Low – Area FPGA Design for the Post – Quantum Cryptography Proposal Round 5, in: Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS), Cryptography and Security Systems, Leipzig, September 2019.

[3] R. J. McEliece, A Public-Key Cryptosystem Based On Algebraic Coding Theory, 1978. DSN Progress Report No. 44 114–116.

[4] V. Ustimenko, On new symbolic key exchange protocols and cryptosystems based on hidden tame homomorphism, Dopovidi 10 (2018) 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, M. Klisowski, On Non-commutative Cryptography with Cubical Multivariate Maps of Predictable Density, in: Intelligent Computing, Springer International Publishing, Cham, 2019, pp. 654–674.

[7] M. Noether, L. Cremona, Mathematische Annalen 59 (1904) 1–19.

[8] R. Wagner, M. R. Magyarik, A Public-Key Cryptosystem Based on the Word N Problem, Advances in Cryptology, in: Proceedings of CRYPTO '84, Santa Barbara, California, USA, 1984, pp. 19–36.

[9] L. Goubin, J. Patarin, Bo-Yin Yang, Multivariate Cryptography, Encyclopedia of Cryptography and Security, 2nd ed., 2011, pp. 824–828.

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

[11] L. Sakalauskas., P. Tvarijonas, A. Raulynaitis, Key Agreement Protocol (KAP) Using Conjugacy and Discrete Logarithm Problema in Group Representation Level, INFORMATICA 8 (2007) 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 17 (2006) 285–289.

[13] D. Kahrobaei, B. Khan, A non-commutative generalization of ElGamal key exchange using polycyclic groups, in: IEEE GLOBECOM 2006 - 2006 Global Telecommunications Conference, 2006 p. 4150920. doi:10.1109/GLOCOM.2006.290.

[14] Cao Zhenfu, New Directions of Modern Cryptography. Boca Raton, CRC Press, Taylor & Francis Group, 2012.
[15] B. Fine, M. Habeeb, D. Kahrobaei, G. Rosenberger, Aspects of Nonabelian Group Based Cryptography: A Survey and Open Problems, arXiv:1103.4093 [cs, math], 2011.
[16] A. Myasnikov, V. Shpilrain, A. Ushakov, Non-commutative Cryptography and Complexity of Group-theoretic Problems, American Mathematical Society, 2011. doi:10.1090/surv/177.
[17] I. Anshel, M. Anshel, D. Goldfeld, An algebraic method for public-key cryptography, Math. Res.Lett. 6 (1999) 287–291.
[18] S. R. Blackburn, S. D. Galbraith, Cryptanalysis of two cryptosystems based on group actions, in: Advances in Cryptology – ASIACRYPT '99, Springer, Berlin, 1999, pp. 52–61.
[19] K. H. Ko, S. J. Lee, J. H. Cheon, J. W. Han, J. Kang, C. Park, New Public-Key Cryptosystem Using Braid Groups, in: Advances in Cryptology — CRYPTO, Berlin, Heidelberg, 2000. pp. 166–183.
[20] G. Maze, C. Monico, J. Rosenthal, Public key cryptography based on semigroup actions. Adv.Math. Commun 1 (2007), 489–507.
[21] P. H. Kropholler, 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 81 (2010) 172–186.
[22] J. A. Lopez Ramos, J. Rosenthal, D. Schipani, R. Schnyder, Group key management based on semigroup actions, Journal of Algebra and its applications 16 (2019).
[23] G. Kumar, H. Saini, Novel Noncommutative Cryptography Scheme Using Extra Special Group, Security and Communication Networks ID 9036382 (2017). doi:10.1155/2017/9036382
[24] V. Ustimenko, On desynchronised multivariate El Gamal algorithm, Cryptology ePrint Archive 712 (2017).
[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 ID 2137561 (2019). doi:10.1155/2019/2137561.
[26] V. Ustimenko, On semigroups of multivariate Cremona transformations and new solutions of Post Quantum Cryptography, Cryptology ePrint Archive 133 (2019).
[27] V. Ustimenko, On new multivariate cryptosystems based on hidden Eulerian equations, Dopov. Nath Acad of Sci 5 (2017) 17–24.
[28] V. Ustimenko, On new multivariate cryptosystems based on hidden Eulerian equations over finite fields, Cryptology ePrint Archive 093 (2017).
[29] A. G. Myasnikov, A. Roman'kov, A linear decomposition attack, Groups Complex. Cryptol. 7 (2015) 81–94.
[30] V. A. Roman'kov, A nonlinear decomposition attack, Groups Complex. Cryptol. 8 (2016) 197–207.
[31] V. Roman'kov, An improved version of the AAG cryptographic protocol, Groups, Complex., Cryptol 11 (2019) 35–42.
[32] A. Ben-Zvi, A. Kalka and B. Tsaban, Cryptanalysis via algebraic span, in: Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, 2018, pp. 255–274.
[33] B. Tsaban, Polynomial-time solutions of computational problems in noncommutative-algebraic cryptography, J. Cryptol 28 (2015) 601–622.