

On the stream ciphers based on the hidden algebraic graphs^{*}

Vasyl Ustimenko^{1,2,†}, Oleksandr Pustovit^{2,*,‡}, and Tymofii Svyrydenko^{2,‡}

¹ Royal Holloway University of London, Egham Hill, Egham TW20 0EX, United Kingdom

² Institute of Telecommunications and Global Information Space, NAS of Ukraine, 13 Chokolivsky ave., 02000 Kyiv, Ukraine

Abstract

Jordan-Gauss graphs are bipartite graphs given by special quadratic equations over the commutative ring K with unity with partition sets K^n and K^m such that the neighbour of each vertex is defined by the system of linear equation given in its row-echelon form. We use special families of such graphs of increasing girth defined over arithmetical rings Z_q , $q=2^m$ for the construction of new graph based stream ciphers of multivariate nature. The degree of multivariate encryption map is not bounded by the constant. So cryptosystem is resistant to Post Quantum adversarial attacks with the interception of many pair of kind plaintext/ciphertext. The speed of encryption is linearly dependent from the growth of the dimension of space of plaintext. It is of size $O(n)$. The cipher has good mixing properties. This is the first family of graph based stream ciphers such that the information on the graph is partially hidden as part of the password. So adversary is unable to use Dijkstra algorithm of finding shortest path for cryptanalytical studies. The algorithm is given with the infrastructure for the key establishment based on ideas of Noncommutative Cryptography and the use of generators of pseudorandom or genuinely random sequences of symbols.

Keywords

Graph based Cryptography, Jordan-Gauss graphs, Graphs of Increasing Girth, Noncommutative Cryptography, Multivariate maps

1. Introduction

Graph based cryptography is rapidly developing now. Papers [1, 2] (see also [3–5]) were the first publications on graph based cryptography. In fact [1] was the first paper on the cryptographical applications of algebraic graphs given defined by the system of equations over commutative ring K . The paper [1] presents generalisations $D(n, K)$ of known algebraic graphs $D(n, q)$, defined over the finite field F_q (see [6] and further references). The properties of cryptographic algorithms based of $D(n, K)$ essentially depend on the choice of finite commutative ring K . The case when K is arithmetical ring Z_q , $q=2^r$, $r>1$ is very convenient for the design of stream ciphers. First implementation of these stream ciphers were described in [6]. We refer to this encryption algorithm as UMCS-cipher.

Some stream ciphers based on graphs given by adjacency matrices can be found in [9–11] or survey [12]. Matrices of graphs also were used for the constructions of key dependent message authentication codes (see [13–15]). Cryptanalytic results on these codes can be found in [16]. Recently new message authentications codes based on graphs $D(n, q)$ were presented [17]. Graphs $D(n, q)$ together with Cayley expanding graphs were used also for the construction of LDPC codes [18–20].

Conventionally the encryption algorithm has to be known and its security rests on the password which correspondents keep in a secure way.

That is why the graph used in UMCS cipher is known. In current paper we use generalisations of graphs $D(n, K)$ introduced in [8]. The equations of these graphs depends on the set S of positive

^{*} CPITS-II 2025: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, October 26, 2025, Kyiv, Ukraine

^{*} Corresponding author.

[†] These authors contributed equally.

✉ vasy1.ustymenko@rhul.ac.uk (V. Ustimenko); sanyk_set@ukr.net (O. Pustovit); ipz23-t.svyrydenko@nubip.edu.ua (T. Svyrydenko)

ORCID 0000-0002-2138-2357 (V. Ustimenko); 0000-0002-3232-1787 (O. Pustovit); 0009-0002-2275-9846 (T. Svyrydenko)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

integers of prescribed cardinality. We present the algorithm based on the graphs where large subset of S is the part of the key. Our task is to preserve good properties of the UMCS cipher such as linear execution speed, good mixing properties, high level of avalanche effect, the change of single character of the plaintext lead to the change of at least 98 percents of the characters in the corresponding ciphertext.

The key parameters of new and old algorithm are subdivided into passive and active passwords which are tuples in the alphabet K . In fact coordinates of the active keys are elements of the multiplicative group K^* of the commutative ring K . The active key is an element of $(K^*)^{l(n)}$, $l(n)=O(n)$ where n is the dimension of the affine space of plaintexts. In fact any parameter $l(n)$, $l(n)<n/4$ can be selected. It means that if the adversary knows plaintext and does not possess an access to plaintexts then in the case of $K=Z_q$ he/she has to use $(q/2)^{l(n)}$ attempts to break the cipher.

The disadvantage of the UMCS stream cipher proposed is the cubic nature of the inverse for the cubic encryption map in many variables. M. Klisowski [21] discovered that the interception of $O(n^3)$ pairs of kind plaintext/ciphertext allows to break the cryptosystem in time $O(n^{10})$.

New cryptosystem is resistant against plaintext /ciphertext attack because of encryption map has degree m , $m>cn$ for some positive constant c .

The introduced cryptosystem allows various obfuscations without the change of mentioned above positive features. One can change the ring Z_q for the general finite commutative ring K . The graph used for encryption possess the homomorphism on graphs $D(m, K)$ for the selected parameter m . This graph can be changed for any *sparse linguistic cover* of $D(m, K)$, i. e. on linguistic graph given by equations with $O(n)$ coefficients such that first $m-1$ of them coincide with the equations of $D(m, K)$. Some obfuscation options are discussed in Section 4.

Section 2 contains definitions of linguistic graphs, Jordan-Gauss graphs and family of graphs $D(n, K)$. The new stream cipher over the ring Z_q , $q=2^r$ is introduced in Section 3.

In Section 5 we discuss the implementation of the algorithm. Conclusive remarks are located in Section 6.

2. On graphs given by triangular equations

Let us assume that K is a commutative ring with unity and ${}^nI(K)={}^nI$ is a bipartite linguistic graph of type $(1, 1, n-1)$, i. e. the incidence structure with the partition sets formed by points $(x)=(x_1, x_2, \dots, x_n)$, $x_i \in K$ and lines $[y_1, y_2, \dots, y_n]$, $y_i \in K$ such that $(x)^nI[y]$ if and only if the following equations hold.

$$\begin{aligned} a_2x_2 - b_2x_2 &= f_2(x_1, y_1), \\ a_3x_3 - b_3x_3 &= f_3(x_1, x_2, y_1, y_2), \end{aligned} \quad (1)$$

...

$$a_nx_n - b_nx_n = f_n(x_1, x_2, \dots, x_{n-1}, y_1, y_2, \dots, y_{n-1})$$

where a_i and b_i are elements of multiplicative group K^* of the K and $f_i \in K[x_1, x_2, \dots, x_{i-1}, y_1, y_2, \dots, y_{i-1}]$.

We use term linguistic graph also for bipartite graphs with infinite tuples of kind $(x_1, x_2, \dots, x_n, \dots)$ as points and infinite tuples $[y_1, y_2, \dots, y_n, \dots]$ defined by infinite system of equations of kind (1).

We define the operator $N_a(v)$ of taking neighbour of colour a via the following rule. If vertex v is the point (p) then $N_a(p)$ is the neighbouring line of this point with the colour a , i. e. line $[l]=[a, l_2, \dots, l_n]$ where l_i , $i=2, 3, \dots, n$ are defined recurrently from the equations $a_2p_2 - b_2l_2 = p_1a$, $a_3p_3 - b_3l_3 = f_3(p_1, p_2, l_1, l_2), \dots$, $a_ip_i - b_il_i = f_i(p_1, p_2, \dots, p_{i-1}, l_1, l_2, \dots, l_{i-1})$. If vertex v is the line $[l]$ then $N_a(l)$ is the neighbouring point of the line with the colour a , i. e. point $(p)=(a, p_2, \dots, p_n)$ where p_i , $i=2, 3, \dots, n$ are defined recurrently from the equations $a_2p_2 - b_2l_2 = al_1$, $a_3p_3 - b_3l_3 = f_3(p_1, p_2, l_1, l_2), \dots$,

$$a_ip_i - b_il_i = f_i(p_1, p_2, \dots, p_{i-1}, l_1, l_2, \dots, l_{i-1}).$$

where a_i and b_i are elements of multiplicative group K^* of the K and $f_i \in K[x_1, x_2, \dots, x_{i-1}, y_1, y_2, \dots, y_{i-1}]$. The first coordinates $q(x) = x_1$ and $q(y) = y_1$ defines colours of the point and line. The walk $v_0lv_1lv_2 \dots v_{k-1}lv_k$ such that colours of all points and colours of all lines are different is the path in the graph formed by distinct vertices.

If $F_n(K)$ is a family of linguistic graphs then the growth of n to infinity defines projective limit $F(K)$. We say that a linguistic graph is of Jordan-Gauss type if the map $[(x), [y]] \rightarrow (f_2(x_1, y_1), f_3(x_1, x_2, y_1, y_2), \dots, f_n(x_1, x_2, \dots, x_{n-1}, y_1, y_2, \dots, y_{n-1}))$ is a bilinear map into K^{n-1} .

Thus, all f_i are special quadratic maps. In the case of Jordan-Gauss graphs, the neighbourhood of each vertex is given by the system of linear equations written in its row-echelon form.

Let ${}^nI(K)$ be a linguistic graph defined over the commutative ring K . For each $b \in K$ and $(p) = (p_1, p_2, \dots, p_n)$, there is the unique neighbour $[l] = N_b(p)$ of the point with the color b . Similarly, for each $c \in K$ and line $l = [l_1, l_2, \dots, l_n]$ there is the unique neighbour $(p) = N_c([l])$ of the line with the color c .

Let $i(1), i(2), \dots, i(k)$ be an increasing sequence of elements from $\{2, 3, \dots, n\}$ and polynomials $g_{i(s)}(x_1, x_2, \dots, x_{i(s)}) \in K[x_1, x_2, \dots, x_{i(s)}]$, $s=1, 2, \dots, k$ such that for each pair of vertexes v and w with coordinates v_1, v_2, \dots, v_n and w_1, w_2, \dots, w_n from the same connected component of I the equalities $g_s(v_1, v_2, \dots, v_{i(s)}) = g_s(w_1, w_2, \dots, w_{i(s)})$ for $s=1, 2, \dots, k$ as family of triangular connectivity invariants of the linguistic graph ${}^nI(K)$ of type $i(1), i(2), \dots, i(k)$.

We say that ${}^{n+m}I(K)$ is linguistic cover of ${}^nI(K)$ if first $n-1$ equations of ${}^{n+m}I(K)$ coincides with the equations of ${}^nI(K)$.

Note that the projection of points and lines of ${}^{n+m}I(K)$ on its first n coordinates defines homomorphism of the graph ${}^{n+m}I(K)$ onto ${}^nI(K)$.

The following statements instantly follows from the definitions.

Proposition 2.1

Let ${}^{n+m}I(K)$ be a linguistic cover of ${}^nI(K)$. Then triangular linguistic connectivity invariants of ${}^nI(K)$ are triangular connectivity invariants of ${}^{n+m}I(K)$.

We refer to the cycle C of linguistic graph nI of type $(1, 1, n-1)$ as multiplicative cycle if for each vertex v of C the difference of colours of its neighbours is an element of multiplicative group K^* of K .

We define multiplicative girth $mug = mug({}^nI(K))$ of ${}^nI(K)$ as minimal length of its multiplicative cycle.

As it follows from the fact that linguistic graphs are bipartite graphs its multiplicative girth is even parameter.

Note that if K is a field then multiplicative girth of linguistic graph I over K coincides with its girth.

We say that the path of vertexes v_1, v_2, \dots, v_s of even length of linguistic graph is multiplicative of length s if

the differences of colours of $v_i - v_{i+2}$ are elements of K^* for $i=0, 1, \dots, s-2$. The linguistic graph I has multiplicative depth $d = mud(I)$ if d is the maximal number such that ends of distinct multiplicative paths started at arbitrary vertex v_0 are distinct.

The multiplicative depth of the graph $D(n, K)$ is at least $\lceil (m+3)/2 \rceil$.

Proposition 2.2. Let us ${}^{n+m}I(K)$ be a linguistic cover of ${}^nI(K)$. Then $mug({}^{n+m}I(K)) \geq mug({}^nI(K))$ and $mud({}^{n+m}I(K)) \geq mud({}^nI(K))$.

3. On the algorithms based on linguistic graphs with increasing multiplicative girth

Assume that $K = \mathbb{Z}_q$, $q = 2^r$. We use the family of linguistic graphs ${}^nI(K)$, $n=2, 3, \dots$ for which $mug({}^nI(K)) \geq m(n)$, where $m(n)$ some increasing function. Assume that $l(n)$ is function with the value of size $O(n)$. For each parameter n the linguistic cover ${}^{n+l(n)}I(K)$ is defined. We assume that for each parameter n the list $g_{i(s)}(x_1, x_2, \dots, x_{i(s)}) \in K[x_1, x_2, \dots, x_{i(s)}]$, $s=1, 2, \dots, k$, $k=k(n)$ of triangular connectivity invariants of ${}^nI(K)$ is known.

We assume that graph ${}^nI(K)$ and its triangular connectivity invariants are known for public, but equations with numbers $n, n+1, \dots, n+l(n)$ of ${}^{n+l(n)}I(K)$ are part of the secret key. So adversary knows just parameter $l(n)$.

The secret key of our stream cipher is the tuples $(\alpha_2, \alpha_3, \dots, \alpha_s)$, $(\gamma_2, \gamma_3, \dots, \gamma_s)$ where $\alpha_j \in (Z_q)^*$ and $(\lambda_2, \lambda_3, \dots, \lambda_s)$ where λ_i are even residues together with the tuple $((a(2), a(3), \dots, a(s)))$ formed by elements of $a(j) \in \{1, 2, \dots, 2^{r-1}-1\}$. Additionally both correspondents know nonlinear polynomials f and h from $K[z_1, z_2, \dots, z_{k(n)}]$.

The active password is formed by *depth parameter* t of size $O(1)$ together with tuples $(\beta(1), \beta(2), \dots, \beta(t))$ and $(\mu(1), \mu(2), \dots, \mu(t))$ from $((Z_q)^*)^m$. We assume that $2t \leq \lfloor m(n)/2 \rfloor - 2$.

Algorithm. Correspondents work with the space of plaintexts K^s , $s = n + l(n)$

The encryption.

We consider bijective transformations L of kind $x_1 \rightarrow x_1 + \alpha_2 x_2 + \dots + \alpha_s x_s$, $x_j \rightarrow x_j$, $j = 2, 3, \dots, s$ and $Q = Q(x_1, x_2, \dots, x_n)$ which sends x_1 to $x_1(\lambda_2 x_2 + \gamma_2)^{a(2)}(\lambda_3 x_3 + \gamma_3)^{a(3)} \dots (\lambda_s x_s + \gamma_s)^{a(s)} = Q$ and do not change x_j , $j = 2, 3, \dots, s$.

Assume that the input is plaintext (x_1, x_2, \dots, x_s) .

Encryption procedure.

Step 1. Compute $Q(x_1, x_2, \dots, x_s) = (z_1, z_2, \dots, z_s) = y(0)$ and treat this vector as a point of graph ${}^s I(K)$. Set $a = z_1$.

Step 2. Compute $g_{i(j)}(y_1, y_2, \dots, y_{i(j)}) = a_{i(j)}$, $j = 1, 2, \dots, k(n)$ and element $(2f(a_{i(1)}, a_{i(2)}, \dots, a_{i(k)}) + 1)y_1 + h(a_{i(1)}, a_{i(2)}, \dots, a_{i(k)}) = c$

and compute $N_c(y_1, y_2, \dots, y_s) = z(0)$

Step 3. Compute $y(1) = N_{a+\beta(1)}(z(0))$ and $z(1) = N_{c+\mu(1)}(y(1))$

Step 4. Compute $y(2) = N_{c+\beta(1)+\beta(2)}(z(1))$ and $z(2) = N_{c+\mu(1)+\mu(2)}(y(2))$

we continue this procedure until step t and get

$y(t) = N_{a+\beta(1)+\beta(2)+\dots+\beta(t)}(z(t-1))$ and $z(t) = N_{c+\mu(1)+\mu(2)+\dots+\mu(t)}(y(t))$

Finally we compute the ciphertext $v = (v_1, v_2, \dots, v_s)$ as $L(y(t))$.

Assume that correspondent get the ciphertext v from his/her partner. The following steps will allow

to restore the plaintext p .

Step 1. The computation of $u = L^{-1}(v) = (u_1, u_2, \dots, u_n, u_{n+1}, \dots, u_s)$.

Step 2. It gives the option to compute the values of triangular connectivity invariants of the graph ${}^n I(K)$ on intermediate point $Q(p) = (z_1, z_2, \dots, z_s)$. Recall that $s = n + l(n)$. We compute list $g_{i(j)}(z_1, z_2, \dots, z_{i(s)}) \in K[z_1, z_2, \dots, z_{i(j)}]$, $j = 1, 2, \dots, k$, $k = k(n)$ as $a_j = g_{i(j)}(u_1, u_2, \dots, u_{i(s)})$, $j = 1, 2, \dots, k$, $k = k(n)$.

Step 3. The computation of $z_1 = a$ as a solution of $(2f(a_{i(1)}, a_{i(2)}, \dots, a_{i(k)}) + 1)z_1 + h(a_{i(1)}, a_{i(2)}, \dots, a_{i(k)}) + \mu(1) + \mu(2) + \dots + \mu(t) = u_1$ for the unknown z_1 . Thus we are able to recover colours of the walk

$a, c = (2f(a_{i(1)}, a_{i(2)}, \dots, a_{i(k)}) + 1)a + h(a_{i(1)}, a_{i(2)}, \dots, a_{i(k)}), a(1) = a + \beta(1), c(1) = c + \mu(1),$
 $a(2) = a(1) + \beta(2), c(2) = c(1) + \mu(2), \dots$

$a(t) = a(t-1) + \beta(t), u_1.$

Step 4. The computation of intermediate vector $(z) = (z_1, z_2, \dots, z_s)$. We compute recurrently $y(t) = N_{a(t)}(u)$,

$Z(t-1) = N_{c(t-1)}(y(t)), y(t-1) = N_{a(t-1)}(z(t-1)), z(t-2) = N_{c(t-2)}(y(t-1)), \dots, z(1) = N_{c(1)}(y(2)), y(1) = N_{a(1)}(z(1)),$

$z(0) = N_c(y(1)), y(0) = N_a(z(0)) = (z_1, z_2, \dots, z_s).$

Step 4. The computation of plaintext x as $Q^{-1}(z_1, z_2, \dots, z_s)$.

Standard obfuscation.

Let E be the described above encryption map on the affine space K^n of plaintexts. We select linear maps L_1 and L_2 of kind $x_i \rightarrow x_i, x_i \rightarrow l_i(x_1, x_2, \dots, x_n)$, $i = 1, 2, \dots, n$ with linear forms l_i of density $O(1)$ such that the inverses of L_i also have density $O(1)$. So we will use $L_1 E L_2$ and $(L_2)^{-1} E^{-1} (L_1)^{-1}$ for the encryption and decryption.

4. On selected families of Jordan-Gauss graphs

The projective limit $D(K)$ of linguistic graphs $D(n, K)$ can be naturally defined via positive roots of the root system $Ext A_1$ (A_1 with wave or extended Dynkin diagram of A_1), see [23]. It is convenient to use positive roots of this root system as indexes of points and lines. The positive real roots of $Ext A_1$ are $(k+1)\alpha_1 + k\alpha_2, k\alpha_1 + (k+1)\alpha_2$ where $k \geq 0$ and α_1, α_2 are simple roots. The system also contains so

called imaginary roots of kind $k\alpha_1 + k\alpha_2$ where $k \geq 1$. We identify roots with their coordinates in the lattice generated by simple roots. Thus we get the list $(k+1, k)$, $(k, k+1)$, $(k+1, k+1)$, $k \geq 0$. We introduce "twins" of imaginary roots denoted as $(k, k)'$ and assume that $(1, 1) = (1, 1)'$. So we get the set of roots R which elements listed as $(1, 0)$, $\{0, 1\}$, $(1, 1)$, $(1, 2)$, $(2, 1)$, $(2, 2)$, $(2, 2)'$, $(2, 3)$, $(3, 2)$, ...

Let us consider affine space of all functions $P = K^{R-[0,1]}$ and $L = K^{R-[1,0]}$ from $R - \{(0, 1)\}$ and $R - \{(1, 0)\}$ to K which have finite supports. We define an incidence structure $D(K)$ with the partition sets P (points) and L (lines) with the incidence relation given by triangular equations. We can assume that points and lines are infinite tuples with coordinates from K defined by indexes from R . The incidence relation I between point $(x) = (x_{10}, x_{11}, x_{12}, x_{21}, x_{22}, x'_{22}, \dots, x'_{ii}, x_{i+1}, x_{i+1,i}, x_{i+1,i+1}, \dots)$ and line $[y] = [y_{01}, y_{11}, y_{12}, y_{21}, y_{22}, y'_{22}, \dots, y'_{ii}, y_{i+1}, y_{i+1,i}, y_{i+1,i+1}, \dots]$ is given by the following equations

$$\begin{aligned} x_{ii} - y_{ii} &= x_{10} y_{i-1,i}; \\ x'_{ii} - y'_{ii} &= x_{i,i-1} y_{01}; \\ x_{i,i+1} - y_{i,i+1} &= x_{ii} y_{01}; \quad (1) \\ x_{i+1,i} - y_{i+1,i} &= x_{10} y'_{ii}. \end{aligned}$$

This four relations are defined for $i \geq 1$ under condition that $x'_{11} = x_{11}$, $y'_{11} = y_{11}$.

Recall that tuples (x) and $[y]$ have finite support, i. e. only finite number of their coordinates differ from zero. Coordinates x'_{ii} and y'_{ii} correspond to the root $(i, i)'$.

As It follows instantly from the results of [22] the infinite Jordan Gauss graphs $D(K)$ does not have multiplicative cycles.

We define graph $D(n, K)$ as the homomorphic image of $D(K)$ defined by the projection of points and lines onto their first n coordinates. So the incidence of Jordan-Gauss graph $D(n, K)$ is defined by the first $n-1$ equations of $D(K)$.

The following statement follows from the results of [22].

Proposition 3. 1. Let K be arbitrary commutative ring. Then $\text{mug}(D(n, K)) \geq n+5$.

To introduce triangular *connectivity invariants* of $D(n, K)$, it will be convenient for us to define $y_{-1,0} = x_{0,-1} = y_{1,0} = x_{l_{0,1}} = 0$, $y_{0,0} = x_{00} = -1$, $y'_{0,0} = x'_{0,0} = -1$.

Graphs $CD(k, K)$ with $k \geq 6$ were introduced in [22] for as induced subgraphs of $D(k, K)$ with vertices u satisfying special equations $a_2(u)=0$, $a_3(u)=0, \dots$, $a_t(u)=0$, $t = [(k+2)/4]$, where $u = (u_{\alpha}, u_{11}, u_{12}, u_{21}, \dots, u_{r,r}, u'_{r,r}, u_{t+1}, u_{r,r+1}, u_{r+1,r}, \dots)$,

$2 \leq r \leq t$, $\alpha \in \{(1, 0), (0, 1)\}$ is a vertex of $D(k, K)$ and

$a_r = a_r(u) = \sum_{i=0,r} (u_{ii} u'_{r-i, r-i} - u_{i,i+1} u_{r-i, r-i-1})$ for every r from the interval $[2, t]$.

for every r from the interval $[2, t]$.

We set $a = a(u) = (a_2, a_3, \dots, a_t)$ and assume that $D(k, K) = CD(k, K)$ if $k=2, 3, 4, 5$.

As it was proven in [22] graphs $D(n, K)$ are edge transitive. So their connected components are isomorphic graphs. Let ${}^v CD(k, K)$ be a solution set of system of equations $a(u) = (v_2, v_3, \dots, v_t) = v$ for certain $v \in K^{t-1}$. Each ${}^v CD(k, K)$

is the disjoint union of some connected components of graph $D(n, K)$. It is known that if K is commutative ring with unity then ${}^v CD(k, K)$ are connected and in the case of $K = F_4$ each ${}^v CD(k, K)$ is union of 4 connected components (see [8] and further references).

The following graphs were introduced in [8]. Let us take graph of kind $D(k+m, K)$ with the triangular connectivity invariants a_j , $j = 2, 3, \dots, [(k+2)/4]$, $[(k+2)/4]+1, \dots, [k+m+2]/4$.

We consider the set ${}^{k+m}S$ of indexes for the coordinates of points and lines of $D(k+m, K)$ with indexes $(i, i)'$ where $i = [(k+2)/4]+1, [(k+2)/4]+2, \dots, [(k+m+2)/4]$ of cardinality $d = [(k+m+2)/4 - [(k+2)/4] + 1$. We select subset \mathcal{J} of ${}^{k+m}S$ and delete coordinates of $D(k+m, K)$ with indexes $(i, i)'$, $j \in \mathcal{J}$ together with the corresponding equations

$x'_{ii} - y'_{ii} = x_{i,i-1} y_{01}$ and change the next equation of (1) for $x_{i+1,i} - y_{i+1,i} = x_{10} y'_{ii}$. We denote obtained graph as ${}^J D_{k,m}(K)$. Note that $D(k, K)$ is the homomorphic image of ${}^J D_{k,m}(K)$. So

$g({}^J D_{k,m}(K))m(K) \geq k+5$. Note that the partition sets of ${}^J D_{k,m}(K)$ are free modules K^s where $s = k+m-|$

5. On the implementation of the algorithm with selected graphs

We have to select finite commutative ring $K=Z_q$, $q=2^r$, $r>2$. So we will work with the space of plaintexts K^r . For the selection of our passive password we present n as $k+m-d$ where k , m and d are linear functions of kind $\alpha n + \beta$, $0 < \alpha < 1$ of size $O(n)$. We will work with the graph ${}^jD_{k,m}(K)$ where $|j|=d$. We assume that m , k , d and the subset J of $\{2, 3, \dots, [(k+2)/4], [(k+2)/4]+1, \dots, [k+m+2]/4\}$ are the part of the passive password. We use invariants a_2, a_3, \dots, a_l , $l=[(k+2)/4]$ of the graph. We select for the passive password the tuples $(\alpha_2, \alpha_3, \dots, \alpha_n)$, $(\gamma_2, \gamma_3, \dots, \gamma_n)$ where $\alpha_j \in (Z_q)^*$, $(\lambda_2, \lambda_3, \dots, \lambda_n)$ where λ_i are even residues together with the tuple $((a(2), a(3), \dots, a(s)))$ formed by elements of $a(j) \in \{1, 2, \dots, 2^{r-1}-1\}$ such that only $O(1)$ of them are different from 1 nonlinear polynomials $f=f(z_1, z_2, \dots, z_l)$ and $h=h(z_1, z_2, \dots, z_l)$ from $K[z_1, z_2, \dots, z_l]$.

We chose f as $(2z_1+1)^{k(1)}(2z_2+1)^{k(2)} \dots (2z_l+1)^{k(l)}$, $k(1)+k(2)+\dots+k(l)=\lfloor n/2 \rfloor$ and g as $c_1z_1 + c_2z_2 + \dots + c_lz_l$.

The active password is formed by *depth parameter* t of size $O(1)$ together with tuples $(\beta(1), \beta(2), \dots, \beta(t))$ and $(\mu(1), \mu(2), \dots, \mu(t))$ from $((Z_q)^*)^m$. We assume that $2t < [(k+5)/2]-2$.

We implemented our algorithms in the case of commutative ring Z_{256} which has the same size with the binary alphabet.

We select the parameters of the passive password as follows. Only $O(1)$ values of kind α_i , β_i , $a(j)$ differs from 1 and only selected finite number of λ_i differs from 2. The passive password remains the same during our computer simulation.

We refer to $d=2m$ as the *length of active password*. High multiplicative girth of the graph insures the following property: change of active password leads to the change of intermediate output $v=(v_1, v_2, \dots, v_n)$.

Computer simulation demonstrate high level of the avalanche effect. We can see that single change of the character of initial file or single change of the character of active password lead to the change of 98% of characters of the ciphertext. Our computations are described in terms of basic operations of the commutative ring $K=Z_{256}$. To speed up the performance of our stream cipher we present the operations of addition and multiplication of the ring via the loaded addition and multiplication tables. So we have embedded functions $x+y$ and $x \boxtimes y$ where x and y are taken from the set of residues mod q . We will use another embedded power function x^y where x is from the domain of all odd residues and y is the residue modulo 2^{m-1} which is the order of multiplicative group K^* . The measurements of the execution time are given in Fig 1.

Our software is written in C++ programming language and therefore it is portable and runs in many platforms such as Unix/Window. To evaluate the performance of our algorithm, we use with different size of files. We measure the time needed to produce the digest in millisecond) and the file size of files in kilobytes for passwords of length d . We use an average PC with processor Pentium 3.00 GHz, 2GB memory RAM and system Windows 7. The following diagram presents the time of the algorithms execution in the case of selected graph as function from size of the file and the length of the active key.

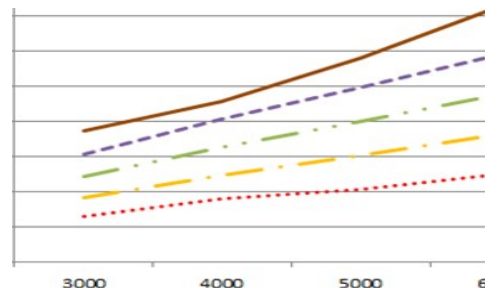


Figure 1: Run time for the digest generation

For the generation of passive key one of correspondents can use the pseudorandom (or random) generator of elements of the arithmetical ring Z_{256} . Currently there are large varieties of generators with the use of different ideas such as classical deterministic methods, use of artificial intelligence

(machine learning, neural networks) or quantum computing. Assume that one (correspondent) generates the pseudorandom sequence (p_1, p_2, \dots, p_n) . Then he/she can form $(\alpha_2, \alpha_3, \dots, \alpha_n)$ (or $(\gamma_2, \gamma_3, \dots, \gamma_n)$) as sequence $(2p_1+1, 2p_2+1, \dots, 2p_{s-1}+1)$, the sequence $(\lambda_2, \lambda_3, \dots, \lambda_n)$ can be formed as

$(2p_1, 2p_2, \dots, 2p_{s-1})$ and (c_1, c_2, \dots, c_l) as (p_1, p_2, \dots, p_l) . The characteristic function of a subset J of S can be formed as $(p_1(\bmod 2), p_2(\bmod 2), \dots, p_k(\bmod 2))$ where k is the cardinality of S .

We refer to the sequences of parameters $\alpha_i, \gamma_i, \lambda_i$ and the set J as pseudorandom data of the key.

The vast majority of parameters $a(2), a(3), \dots, a(s)$ coincides with 1. So correspondent just select few position to input elements of $\{1, 2, \dots, 127\}$ which are differs from 1. He/she also selects integers k_1, k_2, \dots, k_l .

During the communication session protected by the stream cipher correspondent can use pseudorandom sequences of kind (p_1, p_2, \dots, p_n) as plaintexts. So they can safely change pseudorandom data of the key as well as other parameters of the key. Some new algorithms of generating pseudorandom or genuinely random sequences appear recently. They are based on the use of Artificial Intelligence, Machine Learning, Neural Networks and Quantum Computing. (see [32–36]).

In fact computer simulation show us the algorithm is very sensitive to the change of pseudorandom data.

We suggest to use some of the protocols of Noncommutative Cryptography (see [25–27]) implemented with multivariate platforms. We select the implementation [24] for the key establishment before the beginning of communication session. Reader can find some recent cryptanalytic results on Noncommutative Cryptography in [28–30] and [31].

The idea is the following. Correspondents Alice and Bob use the protocol to elaborate nonlinear multivariate map of kind $x_1 \rightarrow f_1(x_1, x_2, \dots, x_s), x_2 \rightarrow f_2(x_1, x_2, \dots, x_s), \dots, x_n \rightarrow f_s(x_1, x_2, \dots, x_s)$, where $f_i, i=1, 2, \dots, s$ are elements of $K[x_1, x_2, \dots, x_s]$. Alice uses the pseudorandom generator to construct sequences $u=(u_1, u_2, \dots, u_s)$ and (p_1, p_2, \dots, p_s) . She sends u to Bob via open channel together with $(p_1+f_1(u_1, u_2, \dots, u_s), p_2+f_2(u_1, u_2, \dots, u_s), \dots, p_s+f_s(u_1, u_2, \dots, u_s))$. He restores (p_1, p_2, \dots, p_s) . Note that $O(1)$ pairs of sequences will allow to make safe delivery of the password from Alice to Bob.

6. Conclusions

We introduce graph based stream cipher which use algebraic graph defined by partially hidden equations. The graph is homomorphic image of well known algebraic bipartite graph $D(k+m, K)$ with the partitions sets isomorphic to the affine space K^m under the homomorphism of deletion of special coordinates indexed by $j, j \geq k+1$. The subset J of cardinality t of coordinates for possible deletion can be arbitrary subset of the set of cardinality $p=[(k+m+2)/4]-[(k+2)/4]$. So we have 2^p options to select S . If n is the dimension of the space of plaintexts then the equality $m+k-t=n$, where m, k are of linear size of kind $\alpha n+b$ with $0<\alpha<1$. The subset S is the part of the key of our stream cipher. So adversary do not know the basic graph in the definition of encryption process.

Note that in all previously known graph based stream ciphers the basic graph was known for the adversary. So he/she could use Dijkstra algorithm of finding the shortest pass for the cryptanalytic investigations of the stream cipher based on walks in the graph.

The idea to use connectivity invariants of graphs $D(n, K)$ for the design of stream cipher was presented in [22] but the previous implementation in the case of $K=\mathbb{Z}_q, q=2^r$ did not use them (see [7], we refer to this cryptosystem as UMCS-cipher). In our new algorithm we preserve the following property of UMCS-cipher. Different active passwords produce distinct corresponding ciphertexts.

It means that in the case when adversary has passive password and does not have access to encrypted data he/she has to complete the brut force search of all $(q/2)^k$ options for possible active password. Note that k is a linear function in variable n of kind $\alpha n+\beta$ with $0<\alpha<1$. So resistance to such attacks by adversary is increasing with the growth of parameter n .

The encryption map of UMCS-cipher as well as its inverse are cubic multivariate cubic transformation. As it was shown in [21] adversary can intercept $O(n^3)$ pairs of kind

plaintext/ciphertext and approximate the encryption map and its inverse in time $O(n^{10})$. The advantage of stream cipher is that the degree of its encryption map is $>2n$ and the inverse is given by rational function. Thus attacks with the interception of plaintexts and corresponding ciphertext are unfeasible.

The algorithm is given with the procedure to change the passive and active key during the communication process.

Declaration on Generative AI

While preparing this work, the authors used the AI programs Grammarly Pro to correct text grammar and Strike Plagiarism to search for possible plagiarism. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the publication's content.

References

- [1] V. Ustimenko, Coordinatisation of Trees and their Quotients, in the Voronoi's Impact on Modern Science, Kiev, Institute of Mathematics, 2 (1998) 125-152.
- [2] V. Ustimenko, Random Walks on Special Graphs and Cryptography, in: AMS Meeting, 1998.
- [3] A. Paszkiewicz, Przykłady zastosowań teorii grafów do konstrukcji szyfrów, in: IV Krajowa Konferencja Zastosowań Enigma'2000, 2000, K-179–183.
- [4] A. Paszkiewicz, Z. Kotulski, K. Kulesza, J. Szczepański, Proposals of Graph-based Ciphers: Theory and Implementations, ResearchGate, 2001.
- [5] Z. Kotulski, et al., Application of Discrete Chaotic Dynamical Systems, DCC method, Int. J. Bifurcation Chaos, 9(6) (1999) 1121–1135.
- [6] F. Lazebnik, V. Ustimenko, A. J. Woldar, New Series of Dense Graphs of High Girth, Bull. Am. Math. Soc. (New Ser.), 32(1) (1995) 73–79. doi:10.1090/S0273-0979-1995-00569-0
- [7] J. Kotorowicz, V. Ustimenko, On the Implementation of Crypt algorithms based on Algebraic Graphs over Some Commutative Rings, Condens. Matter Phys., 11(2(54)) (2008) 347–360.
- [8] T. Chojęcki, G. Erskine, J. Tuite, V. Ustimenko, On Affine Forestry over Integral Domains and Families of Deep Jordan–Gauss Graphs, Eur. J. Math., 11(10) (2025). doi:10.1007/s40879-024-00798-2
- [9] N. K. Geetha, V. Ragavi, Graph Theory Matrix Approach in Cryptography and Network Security, in: 2022 Algorithms, Computing and Mathematics Conf. (ACM), 2022. doi:10.1109/ACM57404.2022.00025
- [10] A. Costache, et al., Ramanujan Graphs in Cryptography, in: Research Directions in Number Theory, Assoc. Women Math. Ser., vol. 19, 2019. doi:10.1007/978-3-030-19478-9_1
- [11] W. M. Al Etaiwi, Encryption Algorithm using Graph Theory, J. Sci. Res. Rep., 3(19) (2014) 2519–2527. doi:10.9734/JSRR/2014/19/004
- [12] P. L. K. Priyadarsini, A Survey on Some Applications of Graph Theory in Cryptography, J. Discrete Math. Sci. Cryptogr., 2013. doi:10.1080/09720529.2013.878819
- [13] J.-P. Tillich, G. Zémor, Collisions for the LPS Expander Graph Hash Function, in: Advances in Cryptology – EUROCRYPT 2008, Lect. Notes Comput. Sci., vol. 4965, 2008, 254–269.
- [14] D. X. Charles, K. E. Lauter, E. Z. Goren, Cryptographic Hash Functions from Expander Graphs, J. Cryptol., 22(1) (2009) 93–113.
- [15] H. Tomkins, M. Nevins, H. Salmasian, New Zémor–Tillich Type Hash Functions over $GL_2(\mathbb{F}_p)$, J. Math. Cryptol., 14(1) (2020) 236–253.
- [16] C. Petit, K. Lauter, J.-J. Quisquater, Full Cryptanalysis of LPS and Morgenstern hash functions, in: Security and Cryptography for Networks, Springer, Berlin, Heidelberg, 2008, 263–277.
- [17] E. Zhupa, M. Polak, Keyed Hash Function from Large Girth Expander Graphs, Albanian J. Math., 16(1) (2022) 25–39.
- [18] P. Guinand, J. Lodge, Tanner Type Codes Arising from Large Girth Graphs, in: Canadian Workshop on Information Theory (CWIT'97), 1997, 5–7.