On quadratic equations of q-regular tree and their applications to symmetric cryptography*

Vasyl Ustimenko[1,2,†], Oleksandr Pustovit[2,*,†], and Tymofii Svyrydenko[2,†]

[1] Royal Holloway University of London, United Kingdom, 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

Graphs $D(n, q)$ and their connected components $CD(n, q)$ were defined 30 years ago. We observe shortly their applications to Extremal, Spectral and Algebraic Graph theory together with applications to Theory of Low Density Parity Check Codes and Symmetric Cryptography. The natural generalizations $D(n, K)$ of $D(n, q)$ are naturally defined over arbitrary commutative ring. We introduce several new flexible stream ciphers based on special walks on these graphs. Their implementations in the case of arithmetical rings Z_q, $q=2^s$ are discussed. Encryption map of two families of ciphers are nonlinear multivariate maps. Users can vary the degree of these maps to control the resistance against attack of the who has some access to unencrypted information, We modified the encryption and decryption of one of the graph based cipher to violate its multivariate nature. Thus adversary could not use instruments of multivariate cryptanalysis for breaking the system.

Keywords

application of graphs of large girth, symmetric cryptography, graph based ciphers, symbolic computations

1. Introduction

Let us consider infinite dihedral group W given by generators s_1, s_2 and relations $(s_1)^2=e$ and $(s_2)^2=e$. Its Weyl geometry is the bipartite graph with partition sets $P=\{g<s_1> \mid g \epsilon W\}$ and $L=\{g<s_2> \mid g \epsilon W\}$ corresponding to binary relation I such that $\alpha I \beta$ if and only if the intersection of $\alpha \epsilon P$ and $\beta \epsilon L$ is nonempty. This graph is the binary tree depicted below on Figure 1

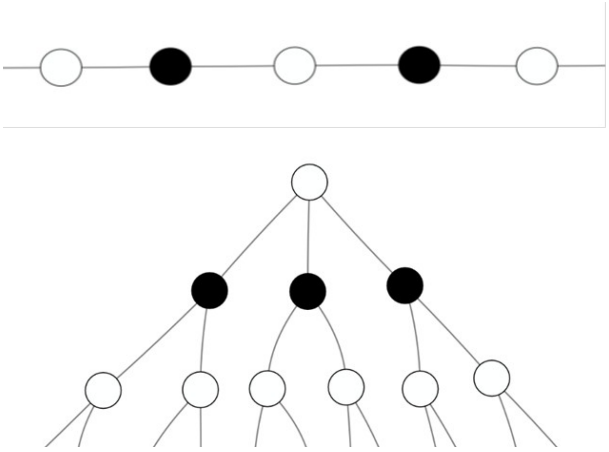

Figure 1: Cubic tree

It is well known that Tits geometry Γ of rank 2 with the flag transitive automorphism group G corresponding to infinite dimensional Lie algebra over finite field F_q with the Weyl group as above is a $q+1$-regular forest (see [1]). Its connected component which is q+1-regular tree is depicted in Fig. 2 for the case $q=2$.

* CPITS 2026: Cybersecurity Providing in Information and Telecommunication Systems, February 28, 2026, Kyiv, Ukraine
* Corresponding author.
† These authors contributed equally.
vasyl.ustymenko@rhul.ac.uk (V. Ustimenko); sanyk_set@ukr.net (O. Pustovt); ipz23-t.svyrydenko@nubip.edu.ua (T. Svyrydenko)
0000-0002-2138-2357 (V. Ustimenko); 0000-0002-3232-1787 (O. Pustovit); 0009-0002-2275-9846 (T. Svyrydenko)

Group G is the BN pair (see [1] or [2]) with the Borel subgroup $B=B^+$ generated by root subgroups corresponding to positive roots. Subgroup B in the ''twinning subgroup '' definition of BN pair can be replaced by ''twinning subgroup '' B^- generated by root subgroups corresponding to negative roots. Then Tits geometry Γ is the $q+1$-regular bipartite graph with partition sets $P' = \{gP_1 | g \in G\}$, $L' = \{gP_2 | g \in G\}$ where $P_i=B^+<s_i>B^+$, $i=1, 2$ are standard maximal parabolic subgroups of G.

In [3] (see also [4]) was noticed that the orbits F_i of B^- containing P_i, $(P_i>B^+)$ are infinite dimensional vector spaces and incidence of element from $F_1 \cap F_2$ can be given by quadratic equations. The incidence structure on these orbits with the set of points F_1, set of lines F_2 and the restriction I' of I on them is a bipartite q -regular forest.

In case of root system $\tilde{A}_1$ coordinates of vector spaces F_i corresponds to negative roots of kind α, β, $k(\alpha)+k(\beta)$, $(k+1)\alpha + k\beta$, $k\alpha + (k+1)\beta$, $k>0$ where α and β are negative simple roots.

So in the basis of α, β indexes presented as elements of the set Ω which consist on $(1, 0)$, $(0, 1)$, (k, k), $(k+1, k)$, $(k, k+1)$, $k>0$. We can assume that points are indexed by elements of $\Omega - \{(0, 1)\}$ and indexes of lines are elements of $\Omega - \{(1, 0)\}$. In fact inexes of kind (k, k) corresponds to imaginary roots of $\tilde{A}_1$ and other indexes correspond to real roots. In [5] authors add twining roots $(i, i)'$ for imaginary roots of kind (k, k) and use extra variables to simplify the quadratic system of equations. It allows them to define an infinite regular forest $D(q)$ with points

$(p_1, p_2, \dots, p_n, \dots)$ and lines

$[l_1, l_2, \dots, l_n, \dots]$

and incidence relation given by the following equations:

$l_2 p_2 = l_1 p_1$,
$l_3 p_3 = l_2 p_1$,
$l_4 p_4 = l_1 p_2$,
$l_i p_i = l_1 p_{i-2}$,
$l_{i+1} p_{i+1} = l_{i-1} p_1$,
$l_{i+2} p_{i+2} = l_i p_1$,
$l_{i+3} p_{i+3} = l_1 p_{i+1}$ where $i \geq 5$.

The quadratic equations of single tree $CD(q)$ of $D(q)$ for odd q can be also written. They are written as conditions on points and lines given via quadratic system of equations in [6].

Let $D(n, q)$ $(CD(n,q))$ be the homomorphic image of $D(q)$ ($CD(q)$ respectively) under the homomorphism of deletion of coordinates of points and lines of $D(q)$ with indexes that $> n$ (see [7]).

During last 30 years many important applications of these graphs to Extremal ([7], [8] and further references), Spectral (see [8] and further references) and Algebraic Graph Theory ([8], [9], [10]), Theory of LDPC Codes (see [11], [12], [13]) and Cryptography (see [14], [15], [16], [8], [10], [17] and further references).

Many of applications are motivated by the fact that the girth of $D(n,q)$ is at least $n+5$. Recall that the girth of simple graph is the length of its minimal cycle. We can take any commutative ring K and consider the graphs D(n, K) obtained via the change of F_q for K (see [14]). Various stream ciphers based on walks on graphs D(n, K) are known since 1998 (see [16], [14], [17] and further references).

Alternative research on symmetric graph based cryptography can be found in [18] – [24]. In paper [14] was proven that in the case of integral domain K the girth of $D(n, K)$ is bounded below by $n+5$. In fact this paper contains the algorithm of construction of enveloping tree of each vertex of $D(n, K)$ of size order $t\,(m)^s$ where $t=|K|$, $s=[[n+5]/2]-1$ where K is finite commutative ring of order t and n is the order of multiplicative group K^* and symbol $[x]$ denotes floor function from rational number x.

We present new stream cipher with the space of plaintexts $(Z_q)^n$ defined over arithmetical ring Z_q of residues modulo q. The parameter q is selected as even prime power 2^s, $s>7$. Additionally we present another stream cipher based on homomorphic image.

So we define our algorithms in terms of $D(n, K)$ for finite commutative ring K.

2. On multivariate transformations of affine spaces Kn and constructions of stream ciphers

Communication Theory widely uses finite commutative rings $(K,+,)$ for the representation of information pieces as tuples of kind $v=(p_1, p_2, ..., p_n) \epsilon K^n$ where coordinates p_i are elements of the alphabet K.

Passwords in form of tuples v are used in symmetric encryption and as access data to use the resources of Information System. Effective encryption maps can be selected as polynomial transformations of affine space K^n.

In case of classical multivariate cryptography developers of cryptosystems concentrated on appications of maps of degree 2 and 3 defined over finite fields (see [30] – [41] and [42] on graph based multivariate public keys) but on case of symmetric cryptography these restrictions are immaterial.

Famous affine Cremona semigroup ${}^nCS(K)$ (see [27] and [28], [29] and further references) is a natural monoid acting on K^n.

Affine Cremona Semigroup ${}^nCS(K)$ is defined as a totallity of endomorphisms of polynomial ring $K[x_1, x_2, \dots , x_n]$. It acts naturally on the affine space K^n as semigroup of polynomial transformation. Element σ of the semigroup can be given via its values on variables, i. e. as the rule $x_i \longrightarrow fi(x_1, x_2, \dots , x_n), i=1, 2,..., n.$

This rule induces the map $\sigma': (a_1, a_2, \dots , a_n) \longrightarrow (f_1(a_1, a_2, ..., a_n), f_2(x_1, x_2, ..., x_n), ..., f_n(x_1, x_2, ..., x_n))$ on the vector space K^n.

We assume that σ is given in its standard form which is the list of fi written as sums of their monomial terms ordered lexicographically. We define the density of fi as the number of their monomial terms and density $den(\sigma)$ of endomorphism σ as maximum of densities of fi. Degrees and densities of multivariate maps make an impact on the complexity of the compositions of two elements.

Computer algebra of multivariate transformations does not include yet effective quantum computations. That is why these area can serve as the source of encryptions tools in Post Quantum Era and Multivariate Cryptography remains one of the fifth directions of Post Quantum Cryptography.

The computation of the composition of two ultivariate maps is very costly because of the product of two elements of degree r and s has degree rs. Recall that degree of σ is the maximum of degrees of fi, It means that n-th degree of nonlinear element usually has exponential size and its computation is unfeasible. Note that for in the case of bijective multivariate transformation F of order d the inverse map is F^{d-1}. Thus the general problem of finding the inverse for F or even finding the reimage of given value b of F are not solvable in polynomial time.

We define *polynomial accelerator* of F as a piece of information T such that its knowledge allows to compute the reimage of map F in polynomial map.

We will use walks on graphs $D(n, K)$ for the construction of polynomial accelerators of polynomil transformations of affine space K^n.

It is well known that applications of studies of lattice or linear codes could lead to valuable cryptographic algorithms. We will use families of graphs with well defined projective limit for the stream ciphers design.

3. Walks on infinite graphs and cubic transformations of affine spaces

Note that quadratic equations (1) of infinite graph $D(K)$ gives the description of neighbourhood of each vertex via the infinite system linear system written in its row echelon form. If we take the point $(c)=(c_1, c_2, ...)$ then its neighbourhood is described by the equations (1) with variables p_i and l_i is specialised as $p_i=c_i$. So the choice of $l_1=a, a \epsilon K$ defines the neighbour $N_a(c)$. We refer to the first coordinate c_1 of point (c) (or line $[c]=[c_1, c_2, ...]$ as the colour of the point (line respectively).

So we have well defined operator $N_a(v)$ of taking the neighbour of v. Let $I=I(K)$ stands for the incidence relation of the bipartite graph $D(K)$. If we take fixed point (u) of colour b and the

sequence $s=(a_1, b_1, a_2, b_2, ..., a_k, b_k)$ of elements K then it defines the walk of the graph of kind $uIv_1Iu_1 Iv_2Iu_2I ... v_kIu_k$ where v_i stands for the point of colour b_i and u_i is the line of colour a_i.

The standard homomorphism $\eta= \eta_n$ of $D(K)$ onto $D(n, K)$ defined via deleting coordinates with indexes $>n$ allows us to take the sequence $\eta(u), \eta(v_1), \eta(u_1), \eta(v_2), \eta (u_2), ... , \eta(v_k), \eta(u_k)$.

Graph $D(K)$ is defined over arbitrary commutative ring K. So we can consider the graph $\check{D}(K)= D(K[x_1, x_2, ... x_n, ...])$ where $x_i, i=1, 2,...$ form an infinite set of unknowns.

If K is an integrity domain, i. e. commutative ring does not contain zero divisors, then both graphs $D(K)$ and $\check{D}(K)$ are forests. Even in the case of finite K the neighbourhood of each vertex of $\check{D}(K)$ is an infinite set.

We consider an induced subraph D' of $\check{D}(K)$ of vertexes with colours from $K[x_1]$. We take some point of D' of kind $(p)=(f(x_1), f_2(x_1, x_2,...), f_3(x_1, x_2, ...),...)$ of colour $f(x_1)$ and sequence w of elements $(A_1(x_1), B_1(x_1), A_2(x_1), B_2(x_1), ..., A_k(x_1), B_k(x_1))$ from $K[x_1]^k$. It defines walk $w \epsilon D'$ with starting point p and colours of other vertexes taken from the sequence $B_1(f(x_1)), A_1(f((x_1), B_2(f(x_1)), A_2(f(x_1))$.

We refer to the sequence w as symbolic word. If we have other symbolic word $w'=(A'_1(x_1), B'_1(x_1), A_2(x_1), B'_2(x_1), ..., A_l(x_1)$ we can take walk w and extend it from its last point ${}^1p$ of colour $B_k(f(x_1)$ (via consecutive adding neighbours with colours $A'_1(B_k(f_1(x_1)), B'_1(f_1(x_1)), A'_2(B_k(f_1(x_1)), B'_2(f_1(x_1)), ...$

This construction motivates the definition of composition of symbolic words w and w' as $(A_1(x_1), B_1(x_1), A_2(x_1), B_2(x_1), ... , A_k(x_1), B_k(x_1). A'_1(B_k(x_1)), B'_1(B_k(x)), A'_2(B_k(x_1)), B'_2(B_k(x_1)), ..., A'_l(B_k(x_1)), B'_l(B_k(x_1)))$.

This operation will define semigroup $SD(K)$. We can see that this is the semidirect product of free semigroup of words in the alphabet $K[x_1]$ and Cremona semigroup ${}^1CS(K)$.

We define homomorphism φ of $SD(K)$ onto $End(K[x_1, x_2....]$ such that $\varphi(w)$ for w = ($A_1, B_1, A_2, B_2, ..., A_k, B_k$) is endomorphism of sending the tuple $(x_1, x_2, x_3, ...)$ to $(u_1, u_2, u_3, ...)$ which is the ending point of the walk of length $2k$ starting from $(x)=(x_1, x_2, ...)$ and containing vertexes with colours $A_1, B_1, A_2, B_2, ..., A_k, B_k$ of the graph $D'(K)$.

The rule $\eta_n(\varphi)$ define the homomorphism φ_n of $SD(k)$ onto ${}^nCS(K)$. Endomorphisms $\varphi(w)$ and $\varphi_n(w)$ are automorphisms if and only if B_k is an automorphism of $K[x_1]$ (see [10] and further references).

Assume that $W(K)= \varphi(CD(K))$ and $W(n, K)= \varphi_n(CD(K)$. For each symbolic word $w=(A_1, B_1, A_2, B_2, ..., A_k, B_k)$ we define w^* as the tuple $(B_k^{-1}(A_k), B_k^{-1}(B_{k-1}), B_k^{-1}(A_{k-1}), ... , B_k^{-1}(B_1), B_k^{-1}(A_1), B_k^{-1})$. We can check that $\varphi(w \cdot w^*)=e$ and $\varphi_n(w\cdot w^*)=e$ for $n \geq 2$.

Let us consider the subsemigroup $S(K)$ of $CD(K)$ formed by symbolic words with coordinates of kind $x_1+c, c \epsilon K$. So $G(K)= \varphi(S(K))$ and $G(n, K)= \varphi_n(S(K))$ are subgroups of $W(K)$ and $W(n, K)$ respectively.

Theorem [25]. Maximal degree of elements of $G(n, K)$ is 3.

We can encode the pair of symbolic words of kind $w=x_1+a_1, x_1+b_1, ..., x_1+a_k, x_1 +b_k) , w^*=(x_1-b_k+a_k, x_1-b_k+b_{k-1}, x_1-b_k+a_{k-1}, ... x_1-b_k+a_1, x_1=b_k)$ as $(a_1, a_2, ..., a_k, b_1, b_2, ... , b_k)$ and $(-b_k +a_k, -b_k+a_{k-1}, ..., -b_k+a_1, -b_k+b_{k-1}, -b_k+b_{k-2}, ... , b_k+b_1, -b_k)$.

Note that $(a_1, a_2, ..., a_k , b_1, b_2, ..., a_k, b_k)\cdot (a'_1, a'_2, ... a'_l, b'_1, b'_2, ... , b'_l) = (a_1, a_2, ..., a_k, b_1, b_2, ... , b_k, -b_k+a'_1=b_k+a'_2, ..., -b_k+a'_l, - b_k+b'_1, -b_k+b'_2, ..., -b_k+b'_l)$. We will use the following statement which follows from results of [14] (3 colours lemma). Let $w=(A_1, B_1, A_2, B_2, ..., A_k, B_k)$ be the symbolic word of $W(K)$. Theh the degree of $\varphi(w)$ is bounded below by $D=1+deg(A_1)+deg(B_1-x_1)+deg(A_2-A_1)+deg(A_3-A_2)+...deg(A_k-A_{k-1})+deg(B_2-B_1)+deg(B_3-B_2)+...+deg (B_k-B_{k-1})$.

Remark. If $n>l$ then $deg (\varphi(w))=deg(\varphi_n(w))$.

Remark. Results of computer simulations support the conjecture that $deg (w)=D+1$.

Let us define regular walk of length k from vertex v as the walk v of kind $vIu_1Iv_1u_2v_2....u_kv_k$ such that differences of colours $c(u_i)-c(u_{i+1})$ $c(v)-c(v_1)$ and $c(v_i)-c(v_{i+1})$ are elements of K^*.

Lemma 1. If v and v_k from $D(K)$ are joint by the regular walk of length $\leq k$ then this walk is uniquely defined.

Lemma 2 If v and v_k from $D(n, K)$ with $k <[n/4]$ are joined by regular walk of length $\leq k$ then this walk is uniquely defined.

Let us consider the homomorphic image $A(K)$ of $D(K)$ defined as bipartite graph with the set of points $(p_1, p_2, ...)$ nd $[l_1, l_2, ...]$ and incidence relation given by equations

$l_2-p_2=l_1p_1$

$l_3-p_3= p_11_2$

$l_4 - p_4=l_1p_3$

l_5-p_5=p_1l_4

…

We define $A(n, K)$ as $\eta_n(A(K))$. In all statements of this section we can exchange $D(K)$ for $A(K)$ and $D(n, K)$ for $A(n, K)$.

4. On the families of stream ciphers

Algorithm 1. Alice and Bob share finite commutative ring K , parameters n and k together with symbolic word $(A_1, B_1, A_2, B_2, …, A_k, B_k)$ with B_k from ${}^1CS(K)$ and two affine transformations L_1 and L_2 from $AGL_n(K)$.

So the encryption map is $E=L_1\varphi_n(w)L_2$. For the decryption they use $E^{-1}=(L_2)^{-1}\varphi(w^*)(L_1)^{-1}$.

Complexity estimates.

Assume that k is of size is $O(1)$ and degrees of $A(i)$ and $B(i)$ are independent from n constants. Alice and Bob do not compute the standard form of E.

The information on L_1, L_2 and symbolic word w allows to compute the cuphertext c of plaintext p via the followings procedure

Step 1. Compute $L_1(p)=v=(d_1, d_2,…, d_n)$ and compute $a_i=A_i(d_1), b_i=B_i(d_1), i=1, 2,…, k$

Step 2. Compute the walk of kind $v, {}^1v, {}^1u, {}^2v, {}^2u, …, {}^kv, {}^ku=u$ in the graph $D(n, K)$ (or $A(n, K)$) such that $r({}^iv)=a_i$, $r({}^iu)=b_i$.

Step 3. Compute ciphertext $c=L_2({}^ku)$.

Notice that the execution of Step 2 takes time $O(n)$. Steps 1 and 2 requires generally $O(n^2)$ elementary operations. If we use affine transformations of kind $x\longrightarrow Mx+b$ where M is sparce matrix with $O(n)$ nonzero entries and b is column tuple from K^n then the execution of Steps 1 and 3 also takes time $O(n)$ as well as computation of the ciphertext.

For the decryption correspondents have to change the symbolic word w for w^* and affine trnsformations L_1 for $(L_2)^{-1}$ and L_2 for $(L_1)^{-1}$ in the described above procedure.

If the matrices of affine transformations $(L_1)^{-1}$ and $(L_2)^{-1}$ are also sparce arrays than the complexity of decryption procedure is $O(n)$.

Remark. The password for this cipher consists on $L_i, i=1, 2$ and symbolic word w.

Modification (Algorithm 2). We introduce the following obfuscation of the algorithm.

After execution of Step 1 and the computation of vertex ${}^ku=u$ we change its colour b_k for $f(d_1)$ where f is a bijective function on K. Finally, correspondent applies L_2 to the vector $(f(d_1), u_2, u_3, …, u_n)$ and gets the ciphertext.

Here we can use $B_k(x_1)$ such that $x_1\longrightarrow B_k(x_1)$ is not a bijection .The decryption procedure is strait forward.

Correspondent applies $(L_2)^{-1}$ to the ciphertext c and gets $(b , u_2, u_3, …, u_n)$ where b is the value of $f(d_1)$. He/she solves the equation $f(x)=b$ for x and gets the solution $x=d_1$ in time $O(1)$. Correspondent restores $u_1=B_k(d_1)$ and colours $a_k=A_k(d_1), b_{k-1}=B_{k-1}, a_{k-1}=A_{k-1}(d_1), …, b_1=B_1(d_1), a_1=A_1(d_1)$.

He/she takes $u={}^ku$, its neighbour ${}^kv$ of colour a_k , neighbour ${}^{k-1}u$ of ${}^kv$ of colour b_{k-1}, …. neighbour of ${}^1u$ of ${}^2v$ of colour b_1, neighbour of ${}^1v$ of ${}^1u$ of colour a_1 and neigbour v of ${}^1v$ of colour d_1.

Finally correspondent restores plaintext as $(L_1)^{-1}(v)$=p.

Remark. The function $\boldsymbol{f}$ is the part of password of Algorithm 2.

Examples. In case of arbitrary commutative ring with unity one can always take f as $ax_1+b, a \epsilon K^*$. In case of $K=F_q$ with even q one can take $f= a(x_1)^2+b$. If $K=Z_q, q=2^s$ we can consider example of $f(x_1)$ which equals $(x_1)^t$ for odd x_1 and equals x_1+b for even x_1 where t is selected odd integer and b is certain even residue.

5. Implementation of the algorithms

Algorithm 5. 1.

We select $K=Z_q, q=2^s, s \geq 8$. So Z^*_q of order 2^{s-1} consists on odd residues. We take parameter $k, k:>0$ and symbolic word w from the subsemigroup $S(Z_q)$ of kind $w=(\alpha_1, \beta_1, \alpha_2, \beta_2, …, \alpha_k, \beta_k)$ where $\alpha_1 \epsilon K, \beta_1 \epsilon K^*, \alpha_{i+1}- \alpha_i \epsilon K^*, \beta_{i+1}- \beta_i \epsilon K^*$

We select intrger n to work with the space of plaintexts K^n and linear transformations L_i such that $L_i(x)=x_1+a_2(i)x_2+a_3(i)x_3+\dots a_n(i)x_n$, $i=1,2$ and $L_i(x_j)=x_j$ for $j>1$. where $a_j(i)$ are elements of K^*.

We refer to the pair k, w as active password and L_i are passive password data. The degree of encryption transformation $E=L_1FL_2$ is 3.

For the computation of homomorphism φ_n we select graph $A(K)$. Under condition that $k\leq[n/4]$ active distinct passwords of kind k, w produce different ciphertext from the selected password. The totality of active passwords with symbolic word of length k has cardinality $q(q/2)^{2k-1}$.

If adversary has no access to unencrypted data he/she has to conduct brut force search via the space of active passwords.

In case of the possibility of interceptions of pairs of kind (plaintext, corresponding ciphertext) adversary has to intercept $O(n^3)$ such pairs and restore the encryption map E in time $O(n^{10})$.

Implementation of the Algorithms 5.1

We provide a direct implementation of the proposed encryption scheme using the ring $K=Z_{256}$. This choice allows us to utilize the standard 8-bit unsigned integer type (`uint8`), ensuring computational efficiency and compatibility with standard byte-level data processing.

Data Structures and Parameters The state of the plaintext is represented as a vector K^n. In our implementation, we utilize `numpy` arrays for vectorized operations. The modulus $q=2^8=256$ corresponds to the natural overflow behavior of the `uint8` data type, which avoids the need for expensive explicit modulo operations in the core loops.

The Encryption Map The core of the algorithm is the neighbor-finding function on the graph $A(K)$. The theoretical relation between a node x (current state) and its neighbor y (next state) given a parameter α (edge label) is defined by a system of equations. In the provided Python code, the function `find_neighbors_v5` implements the edge transition map F. For a given input vector `point_in` (x) and parameter `a` (α), it computes the output `point_out` (y) as follows:

First Coordinate: $y_1=x_1+\alpha(modq)$.

Propagation: For indices $i>1$, the value y_i is derived recursively to satisfy the graph's incidence relations:

1. For even i: $y_i=x_i-y_1x_{i-1}$
2. For odd i: $y+i=x_i-x_1y_{i-1}$

Password and Symbolic Word The "symbolic word" w described in Algorithm 5.1 corresponds to the sequence of parameters used to traverse the graph. In our implementation, `randomize_d_mod` generates a dynamic permutation of the operational space based on a seed, effectively constructing a unique walk path (the active password) for each session.

5.1. Testing and Performance Analysis

To validate the correctness and cryptographic strength of the implementation, we developed a suite of unit tests using the Python unittest framework. The testing strategy focused on: algorithmic consistency (lossless decryption), sensitivity to input changes (avalanche effect), and robustness against noise injection.

Verification of Correctness As demonstrated in the `test_encode_decode_consistency` method, the system successfully restores the original byte array from the ciphertext in all test cases, confirming the algebraic integrity of the inverse map.

Sensitivity and the Avalanche Effect A critical measure of encryption security is the avalanche effect: a slight alteration in the plaintext should result in a significant and unpredictable change in the ciphertext.

We measured this by flipping a single byte at various positions (1/4, 1/2, 3/4 marks) in a 31MB video file and comparing the resulting ciphertext to the original encryption.

1. **Static Path (Passive only):** When using a static graph traversal (fixed parameters), the change propagation was localized, resulting in a "sameness" of ≈99.9%.
2. **Active Password (Randomized Path):** When the "active password" mechanism (randomize_d_mod) was enabled, changing a single byte resulted in a "sameness" of only

0.80%. This indicates that ≈99.2% of the ciphertext bytes were altered, demonstrating lobal diffusion properties.

Noise Injection and Digest Uniqueness To test the system's resistance to differential attacks, we analyzed the impact of noise injection. We generated digests for a 100MB CSV file with a 5% noise ratio. The results showed that even with identical seeds, the introduction of stochastic noise resulted in distinct digests (0.12% difference), ensuring that identical plaintexts do not yield identical ciphertexts when noise is applied.

Performance Analysis We measured the execution time for both encryption and decryption processes across a diverse set of file types to evaluate the algorithm's scalability. The algorithm demonstrates linear scaling O(n).

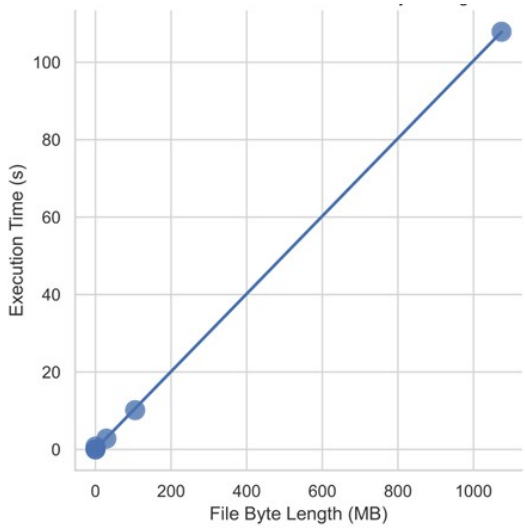

Figure 2: Encryption performance

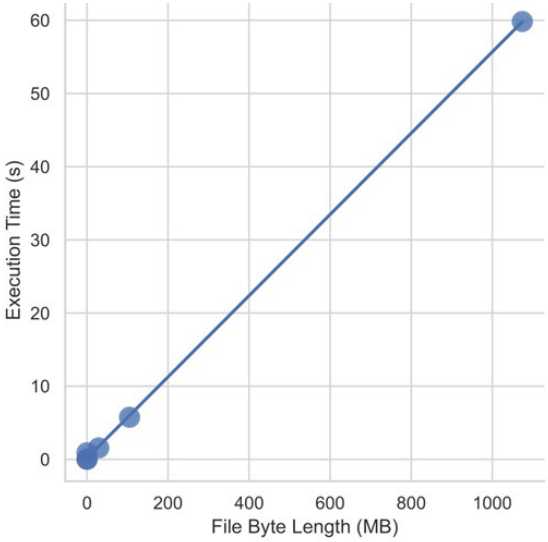

Figure 3: Decryption performance

Algorithm 5.2.

We select the same data with the case of previous algorithm and the tuple of kind $d=(2f_1, 2g_1, 2f_2, 2g_2, \ldots, 2fk, 2g_k)$ where $fi \epsilon K[x_1]$, $i=1, 2, \ldots, k$, $g_i \epsilon K[x_1]$, $i=1, 2, \ldots, k-1$ are polynomials of constant degree $\geq t>1$ such that $deg(fi-fi_{+1}) \geq t$, $deg(g_i-g_{i+1}) \geq t$ for $g_k=\Upsilon x_1$, $\Upsilon \epsilon K^*$.

We form symbolic word w' as the sum of tuple d with the tuple $w=(\alpha_1+x_1, \beta_1+x_1. \alpha_2+x_1, \beta_2+x_1, \ldots, \alpha_k+x_1, \beta_k+x_1)$ of $W(K)$ as in the previous algorithm.

Note that the degree of the map $\varphi_n(w')$ will be at least $1+2kt$ (Proposition 1 for the case of graph $A(K)$).

So correspondents will use encryption map $E=L_1 \varphi_n(w')L_2$, $w' \epsilon W(K)$ and decryption

map $D=L_2^{-1}\varphi_n((w')^*)L_1^{-1}$. The theoretical complexity of the encryption or decryption procedure is $O(n)$. The active password as in the previous algorithm is tuple w. As in previous case each change of the characters of w leads to the change of ciphertext.

The attacks with interceptions of pairs of kind plaintext / corresponding ciphertext are unfeasible because of high degree of encryption and decryption procedure.

Algorithm 5. 3.

We obfuscate the Algorithm 5. 2 accordingly Algorithm 2 of Section 4. For the implementation we select f as in Example of Section 4 for $K=Z_q$. The value of f on odd residue is x^r, where positive odd integer r is taken from open interval $(1, 2^{s-1})$ and the value of f on even residue x is $x+2b$, $b \epsilon K$.

This modification preserves all mentioned above properties of Algorithm 5.2. Encryption and decryption maps of Algorithm 3 are not polynomial transformations. Thus adversary can not use instruments of multivariate cryptanalysis to break the cryptosystem.

6. Conclusions

Graphs D(n, K) defined over finite commutative ring were used for the design of stream ciphers in the paper [16]. This is one of the first publication on graph based Cryptography (see also [24]).

Graphs D(n, K) are defined by equations, it was the First application of algebraic graphs to Symmetric Cryptography. In many cryptographic graph applications the adjacency matrices are used. Various stream ciphers defined in terms D(n, K) and their special homomorphic images are shortly observed in with speed O(n) (see recent paper [17] and further references).

The properties of these algorithms are strongly depend on 7the choice of commutative ring. In the case of K=Zq, q=2s the fastest known algorithm is suggested in [26]. The space of passwords of these even cipher formed by tuples of characters from Zq of length k where k < [(n+5)/2]. The case when distinct passwords produce same ciphertext from seleced plaintext are rare but possible. This algorithm can be modified via the change of the graph A(n, Zq). This modification does not eliminate the collision of passwords.

In Algorithm 5.1 of the paper we change the space of passwords pf stream cipher C from [26] for the set of tuples of special kind. This modification eliminates the case of two passwords producing the same ciphertexr. So we can easily estimate complexity of adversarial attacks under condition that adversary does not have access to unencrypted case. Algorithm 5. 1 is also implemented in cases of graphs D(v, Zq) and A(n,zq). Cipher C and Algorithm 1 have many common properties. The encryption and decryprion procedures correspond to cubic multivariate maps, Both algorithms have good mixing properties.

Algorithm 5.2 poses the mentioned above properties of Algorithm 1 but the degree of multivariate encryption map is linearly growing with the increase of n. So attacks of adversary with the interception of plaintexts with corresponding ciphertexts for map approximations are unfeasible.

Algorithm 5. 3 also has the property to have distinct ciphertexts for different passwords, good mixing properties but its decryption and decryption maps are not multivariate transformations. Thus standard cryptanalytic instruments of Computer Algebra are not applicable in this case.

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] F. Buekenhout, Handbook on Incidence Geometry, North Holland, Amsterdam, 1995

[2] N. Bourbaki, Lie Groups and Lie Algebras, Chapters 1–9, Springer, 1998-2008

[3] V. Ustimenko, On Some Properties of Chevalley Groups and Their Generalisations, In: Investigations in Algebraic Theory of Combinatorial objects, Inst. of System Studies, 1985, 134 –138. Engl. trans.: Kluwer, Dordrecht, 1992, pp. 112–119
[4] V. Ustimenko. On the Embeddings of Some Geometries and Flag Systems in Lie Algebras and Superalgebras, in Root Systems, Representations and Geometries, Kiev, IM AN UkrSSR, pp. 3–16, 1990
[5] F. Lazebnik, V. Ustimenko, Explicit Construction of Graphs with Arbitrary Large Girth and of Large Size, Discrete Applied Mathematics 60, 275–284, 1995
[6] F. Lazebnik, V. Ustimenko, A. Woldar, A Characterization of the Components of the Graphs D(k, q). Discrete Mathematics Vol. 157, Issues 1–3, 1 October 1996, p. 271–283
[7] F. Lazebnik, V. Ustimenko, A.J.Woldar, A New Series of Dense Graphs of High Girth, Bulletin of the AMS 32 (1), p. 73-79, 1995
[8] T. Chojecki, G. Erskine, J. Tuite, V. Ustimenko On Affine Forestry Over Integral Domains and Families of Deep Jordan–Gauss Graphs. Eur. Journal of Mathematics 11, 10 (2025), doi: 10.1007/s40879-024-00798-2
[9] T. Shaska, V. Ustimenko, On the Homogeneous Algebraic Graphs of Large Girth and Their Applications. Linear Algebra and its Applications Vol. 430, Issue 7, 1 April 2009, p. 1826–1837
[10] V. Ustimenko, Graphs in Terms of Algebraic Geometry, Symbolic Computations and Secure Communications in Post-Quantum World, UMCS Editorial House, Lublin, 2022, 198
[11] P. Guinand, J. Lodge, Tanner Type Codes Arising from Large Girth Graphs, Canadian Workshop on Information Theory CWIT'97, Toronto, Ontario, Canada, June 3-6 (1997): LDPC codes based on algebraic graphs.
[12] P. Guinand, J. Lodge, Graph Theoretic Construction of Generalized Product Codes, IEEE International Symposium on Information Theory ISIT'97 Ulm, Germany June 29–July 4 (1997), doi:10.1109/ISIT.1997.613026
[13] D. MacKay and M. Postol, Weakness of Margulis and Ramanujan Margulis Low Density Parity Check Codes, Electronic Notes in Theoretical Computer Science, 74, 8 pp., 2003
[14] V. Ustimenko, On Linguistic Dynamical Systems, Families of Graphs of Large Girth, and Cryptography. J Math Sci 140, 461–471 (2007), doi: 10.1007/s10958-007-0453-2
[15] M. Polak, E. Zhupa, Keyed Hash Function from Large Girth Expander Graphs, Albanian J. Math., 2022, V. 16, No 1 25–39
[16] V. Ustimenko, Coordinatisation of Trees and their Quotients, in the Voronoi's Impact on Modern Science, Kiev, Inst. of Mathematics, 1998, vol. 2, 125–152
[17] V. Ustimenko, O. Pustovit, T. Svyrydenko, On the Stream Ciphers Based on the Hidden Algebraic Graphs, CPITS 2-2025, Kyiv, pp. 358–365
[18] A. Paszkiewicz, Z. Kotulski, K. Kulesza, J. Szczepanski, Proposals of Graph Based Ciphers Theory and Implementations. Research Gate, 2001
[19] A. Costache, B. Feigon, K. Lauter, M. Massierer, A. Puskas, Ramanujan Graphs in Cryptography. In: Balakrishnan, J., Folsom, A., Lalin, M., Manes, M. (eds), Research Directions in Number Theory. Association for Women in Mathematics Series, vol 19
[20] P. L. K. Priyadarsini, A Survey on Some Applications of Graph Theory in Cryptography, J. Discrete Math. Sci. Cryptography, doi:10.1080/09720529.2013.878819
[21] W. M. Al Etaiwi, Encryption Algorithm Using Graph Theory, Journal of Scientific Research & Reports, 3(19): 2519-2527, 2014
[22] X. Denis Charles, Z. Eyal Goren, E. Kristin Lauter., Cryptographic Pash Aunctions from Expander Graphs. Journal of Cryptology, 2009, V. 22, 93–113
[23] H Tomkins, M. Nevins, H. Salmasian New Zemor-Tillich Type Hash Functions Over GL2(F). J. Math. Cryptol.,2020, V. 32, No. 1. 236–253
[24] Z. Kotulski, J. Szczepański, K. Górski, A. Paszkiewicz, & A. Zugaj, Application of Discrete Chaotic Dynamical Systems in Cryptography – DCC method, Int. J. Bifurcation and Chaos, 1999, 9, 1121–1135

[25] A. Wróblewska, On Some Properties of Graph Based Public Keys, Albanian Journal of Mathematics, Vol. 2, 3, 2008, 229–234, Proceedings of NATO Advanced Studies Inst.: New challenges in digital communications
[26] S. J. Kotorowicz, V. A. Ustimenko, On the Implement. of Cryptoalgorithms Based on Algebraic Graphs Over Some Commutative Rings, Condenced Matters Physics, Special Issue: proceedings of the international conferences on finite particle systems, Complex systems theory and its application, Kazimerz Dolny, Poland, 2006, 11 no. 2 (54), 2008, 347–360
[27] M. Noether, Luigi Cremona, Mathematische Annalen, 59 (1904), pp. 1–19
[28] I. R. Shafarevich, On Some Infinite Dimension Groups II, Izv. Akad. Sci. Ser. Math. 2 (1) (1981), 214–226
[29] Y. Bodnarchuk, Every Regular Automorphism of the Affine Cremona Group is Inner, Journal of Pure and Applied Algebra 157 (2001) 115–119
[30] D. Smith-Tone: 2F–A New Method for Constructing Efficient Multivar. Encryption Schemes. In: PQCrypto 2022: The Thirteenth International Conference on Post-Quantum Cryptography, virtual, DC, US, 2022
[31] D. Jayashree, R. Dutta: Progress in Multivar. Cryptography: Systematic Review, Challenges, and Research Directions. ACM Computing Survey, vol. 55, issue 12, No. 246, pp. 1–34, 2023
[32] J. Chen, J. Ning, J. Ling, T. S. C. Lau, Y. Wang, A New Encryption Scheme for Multivar. Quadratic Systems. Theoretical Computer Science, vol. 809, pp. 372–383, 2020
[33] M.-S. Chen, A. Hülsing, J. Rijneveld, S. Samardjiska, P. Schwabe, MQ-Based Signatures in the QROM. In: Proceedings of the IACR International Workshop on Public Key Cryptography, pp. 3–33. Springer, Heidelberg, 2018
[34] J. Ding, A. Petzoldt, D. S. Schmidt, Multivar. Public Key Cryptosystems. 2nd edn. Advances in Information Security. Springer, Heidelberg, 2020
[35] D. H. Duong, H. T. N. Tran, W. Susilo, L. V. Luyen, An Efficient Multivar. Threshold Ring Signature Scheme. Computer Standards & Interfaces, vol. 74, 2021
[36] J.-C. Faugère, G. Macario-Rat, J. Patarin, L. Perret, A New Perturbation for Multivar. Public Key Schemes Such as HFE and UOV. Cryptology ePrint Archive, 2022
[37] W. Buellens, Improved Cryptanalysis of UOV and Rainbow Improved Cryptanalysis of UOV and Rainbow, In: Canteaut, A., Standaert, FX. (eds) Advances in Cryptology – EUROCRYPT 2021. Lecture Notes in Computer Science, vol 12696. Springer, Cham
[38] A. Canteaut, F.-X. Standaert, 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia,, Proceedings, Part I. Springer, Heidelberg, vol. 12696, October 17–21, 2021
[39] M. . Saarinen, D. T.Smith, Post Quantum Cryptography, 15th International Workshop, PQCrypto, Oxford, UK, June 12–14, 2024, Proceedings, Part 2. Springer, Heidelberg, 2024
[40] T. Takagi, M. Wakayama, K. Tanaka, N. Kunihiro, K. Kimoto, Y. Ikematsu, International Symposium on Mathematics, Quantum Theory, and Cryptography, Proceedings of MQC 2019. Open Access, Springer, Heidelberg, 2021
[41] NIST PQC Digital Signature Project: TUOV Specification. https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/spec-files/TUOV-spec-web.pdf
[42] V. Ustimenko, On Schubert Cells of Lie Geometries and Public Keys of Multivar. Cryptography, Contemporary Mathematics, vol. 830, 2025, doi: 10.1090/conm/830/16568