

Article

On Quadratic Equations of the q -Regular Tree and Their Applications in Graph Theory and Cryptography

Vasyl Ustimenko ¹  and Tymoteusz Chojecki ^{2,*} ¹ Department of Mathematics, Royal Holloway University of London, Egham Hill, Egham TW20 0EX, Surrey, UK; vasyi.ustymenko@rhul.ac.uk² Faculty of Mathematics, Physics and Computer Science, Maria Curie-Skłodowska University, ul. Radziszewskiego 10, 20-031 Lublin, Poland

* Correspondence: tymoteusz.chojecki@umcs.pl

Abstract

Graphs $D(n, q)$ and their connected components $CD(n, q)$ were defined 30 years ago. We briefly review their applications to Extremal Graph Theory, Spectral Graph Theory, Algebraic Graph Theory, Symmetric Cryptography, and Theory of Low Density Parity Check Codes. We introduce several new algorithms of Noncommutative Cryptography based on these graphs of large girth. In particular we propose a modification of the Diffie–Hellman protocol in terms of the semigroup of walks of even length on the forest obtained as the projective limit of $D(n, q)$ and the homomorphic image of this monoid, acting on the vector space $(F_q)^n$ as the transformation group $G(n, q)$ of cubic polynomial transformations. The protocol allows users to compute a collision vector from $(F_q)^n$ in time $O(n^2)$. The security of these schemes rests on the complexity of the Conjugacy Power Problem for the affine Cremona semigroup of automorphisms of $F_q[x_1, x_2, \dots, x_n]$. An inverse protocol of El Gamal type allows one to use this scheme for encryption or the creation of digital signatures. Several obfuscations of these algorithms are given.

Keywords: application of graphs of large girth; noncommutative cryptography; graph based key exchange protocols; symbolic computations

MSC: 05D99; 11T71

1. Introduction

Finite geometries associated with finite groups of Lie type and Coxeter groups have traditionally been used in Coding Theory (see [1–5]) and they have also proved useful in Cryptography [6]. One direction of their application is based on the idea of using the largest orbits of the Borel subgroup acting on the geometry and the restriction of the incidence relation to their union (see [7]). This approach has also been used in the case of infinite Tits geometries of rank 2. Below we recall the definitions related to these combinatorial objects.

An *incidence system* is a triple (Γ, I, t) where I is a symmetric irreflexive relation (simple graph) on the vertex set Γ , and $t : \Gamma \rightarrow N$ is a type function onto a set of types N such that $\alpha I \beta$ and $t(\alpha) = t(\beta)$ imply $\alpha = \beta$. An important example of such an incidence system is the so-called *group incidence system* $\Gamma(G, G_s)$, $s \in S$. Here, G is an abstract group and G_s , $s \in S$ are a family of distinct subgroups of G . The elements of $(\Gamma(G), G_s)$, $s \in S$ are the left cosets of G_s in G for all $s \in S$. Cosets α and β are incident precisely when $|\alpha \cap \beta| \neq 0$. The type function is defined by $t(\alpha) = s$, where $\alpha = gG_s$ for some $s \in S$ evaluated to these classical objects (see [8]).



Academic Editors: Iliya Bouyukliev and Kittikorn Nakprasit

Received: 20 April 2026

Revised: 29 June 2026

Accepted: 3 July 2026

Published: 10 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

EXAMPLE 1. Coxeter geometries. Let (W, S) be a Coxeter system, i.e., W is a group with a set of distinguished generators given by $S = \{s_1, s_2, \dots, s_l\}$ and generic relation $(s_i s_j)^{m(i,j)} = e$. Here, $M = (m(i, j))$ is a symmetric l times l matrix with $m(i, i) = 1$ and off-diagonal entries satisfying $m(i, j) \geq 2$. Let $W_i = \langle S_{s_i} \rangle$, $1 \leq i \leq l$ and consider a group incidence system $\Gamma(W) = \Gamma(W, W_i)$, $1 \leq i \leq l$. This geometry is called the Coxeter geometry of W . Then W_i are referred to as the maximal standard subgroups of W .

EXAMPLE 2. Geometries of BN pairs. Let G be a group, B and N be subgroups of G , and S be a collection of cosets of $B \cap N$ in N . We call (G, B, N, S) a *Tits system* (or say that G has a BN -pair) if

- (i) $G = \langle B, N \rangle$ and $B \cap N$ is normal in N ;
- (ii) S is a set of involutions which generate $W = N/(B \cap N)$;
- (iii) sBw is a subset of $BuB \cap BswB$ for any $s \in S$ and $w \in W$;
- (iv) $sBs \neq B$ for all s from S .

Properties (i)–(iv) imply that (W, S) is a Coxeter system (see [9]). Whenever (G, B, N, S) is a Tits system, we call W the Weyl group of the system or more usually the Weyl group of G . The subgroups P_i of G defined by BW_iB are called the *standard maximal parabolic subgroups* of G .

The group incidence system $\Gamma(G) = \Gamma(G, P_i)$, $1 \leq i \leq l$ is commonly referred to as the geometry of BN pairs of spherical type or Tits geometry. The classification of these objects in the case of a finite Coxeter group, i.e., BN pairs of spherical type can be found in [10]. Tits geometries of spherical type coincide with the finite geometries associated with simple groups of Lie type, which are obtained via the groups of inner automorphisms of simple Lie algebras (case of group of normal type or via the groups of fixed points of special symmetries in the group of internal automorphisms (case of groups of twisted type (see [11])).

In these cases, the Coxeter group is interpreted as a group generated by reflections in Euclidean space, i.e., it is a Weyl group defined via the finite root system. Many examples of BN pairs with infinite Coxeter groups are obtained in terms of infinite root systems and internal automorphisms of Kac–Moody Lie geometries [12], [13]. Corresponding BN pairs can be divided into the group of internal automorphisms of related Lie algebras and those constructed via special symmetries of the Weyl group combined with the field automorphism of the same order. Thus we may use the terms “groups of normal type” and “twisted Lie groups” in the case of infinite Weyl group.

Assume that the Weyl group is finite. Group B acts naturally on elements $\Gamma(G)$ and $\Gamma(G)^2$. Orbits of B , $\Gamma(G)^2$ are known as orbitals of B . Orbits of G are referred to as Schubert cells (see [14], [15] and further references). It is known that Schubert cells are vector spaces over the ground field of finite Tits geometry. The restriction of the incidence relation on two Schubert cells is a bipartite graph or empty relation. We can refer to such nontrivial graphs as Schubert orbitals. These orbitals were investigated in [16], [17], and [18]. It was noticed that, in special bases of partition sets, the incidence of a Schubert orbital can be described via a system of triangular quadratic equations, and the neighbourhood of each vertex is described by a system of linear equations written in row echelon form. In the case of Lie groups of normal type, the Chevalley basis of the corresponding Lie algebra ([19], [20]) was used.

In the case of an infinite BN pair corresponding to Kac–Moody algebra with infinite Weyl group, the Borel subgroup $B = B^+$ containing all root subgroups for positive roots is not conjugate to the group B^- containing all root subgroups of negative roots, and the Weyl group is divided on real and imaginary roots [13]. For this reason, there are several ways to define Schubert cells and Schubert orbitals (see [17]). One of these options in the case of rank 2 and root system $\tilde{A}_1$ is considered in [21].

In this case when the Weyl group W is a dihedral group generated by s_1, s_2 and relations $s_1^2 = e$ and $s_2^2 = e$, its Weyl geometry is the bipartite graph with bipartition sets $P = \{g\langle s_1 \rangle | g \in W\}$ (white vertices in Figure 1) and $L = \{g\langle s_2 \rangle | g \in W\}$ (black vertices in Figure 1), corresponding to the binary relation I such that $\alpha I \beta$ if and only if the intersection of $\alpha \in P$ and $\beta \in L$ is nonempty. This graph is the binary tree depicted in Figure 1.

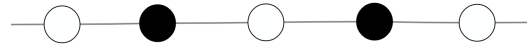


Figure 1. Binary tree.

It is known that the Tits geometry Γ of the adjoint group G of an infinite dimensional Kac–Moody algebra with the Weyl group described above, defined over the finite field F_q , is a $q + 1$ -regular forest (see [8] and Figure 2 for $q = 2$). Let B^+ (B^-) be the Borel subgroups containing the root subgroups corresponding to positive (negative) roots. Then the Tits geometry Γ is a $q + 1$ -regular bipartite graph with partition sets $P' = \{gP_1 | g \in G\}$, $L' = \{gP_2 | g \in G\}$, where $P_i = B^+ \langle s_i \rangle B^+$, $i = 1, 2$ are standard maximal parabolic subgroups of G .

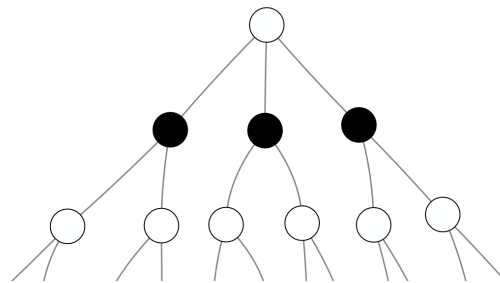


Figure 2. Cubic tree.

It was noticed in [16] (see also [16], [21]) that the orbits F_i of B^- containing P_i are infinite dimensional vector spaces and the incidence relation between elements of $F_1 \cup F_2$ can be given by quadratic equations. The incidence structure on them, with the set of points F_1 , the set of lines F_2 , and the restriction I' of I on them is a bipartite q -regular forest.

In the case of root system $\tilde{A}_1$, coordinates of the vector spaces F_i correspond to negative roots of the form $\alpha, \beta, k(\alpha) + k(\beta), (k + 1)\alpha + k\beta, k\alpha + (k + 1)\beta, k \geq 1$, where α and β are negative simple roots. So in the basis of α, β , the indices are represented by elements of the set Ω which consists of $(1, 0), (0, 1), (k, k), (k + 1, k), (k, k + 1), k \geq 1$. We can assume that points are indexed by elements of $\Omega - \{(0, 1)\}$ and indices of lines are elements of $\Omega - \{(1, 0)\}$. In fact indices of the form (k, k) correspond to imaginary roots of $\tilde{A}_1$ and other indices correspond to real roots. In [22], the authors add twining roots $(i, i)'$ for imaginary roots of kind (k, k) and use extra variables to simplify the quadratic system of equations. This 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 the incidence relation is given by the following equations:

$$\begin{aligned}
 p_2 - l_2 &= l_1 p_1, \\
 p_3 - l_3 &= l_1 p_2, \\
 p_4 - l_4 &= p_1 l_2, \\
 p_i - l_i &= p_1 l_{i-2}, \\
 p_{i+1} - l_{i+1} &= p_{i-2} l_1, \\
 p_{i+2} - l_{i+2} &= p_i l_1, \\
 p_{i+3} - l_{i+3} &= p_1 l_{i+1},
 \end{aligned} \tag{1}$$

where $i \geq 5$, $i \bmod 4 = 1$

The quadratic equations defining a single tree $CD(q)$ of $D(q)$ for odd q can be also written explicitly. They are given as conditions on points and lines via a quadratic system of equations in [23].

Let $D(n, q)$ ($CD(n, q)$) be the homomorphic image of $D(q)$ ($CD(q)$, respectively) under the homomorphism obtained by deleting the coordinates of points and lines of $D(q)$, whose indices are greater than n (see [24]). During the last 30 years, many important applications of these graphs have been found in Extremal Graph Theory ([24], [25], and further references), Spectral Graph Theory (see [25] and further references) and Algebraic Graph Theory ([25], [26]), Theory of LDPC Codes (see [27], [28], [29]) and Cryptography (see [30], [31], [25], [26], and further references). In this paper we present new examples of such applications—new efficient key exchange protocols of Noncommutative Cryptography and El Gamal-type cryptosystems of Postquantum Cryptography to encrypt plaintexts with complexity $O(n^2)$. Many of the applications are motivated by the fact that the girth of $D(n, q)$ is at least $n + 5$. Recall that the girth of a simple graph is the length of its minimal cycle. We can take any commutative ring K and consider the graphs $D(n, K)$ obtained by replacing F_q with K (see [30]). Alternative research on graph-based symmetric cryptography can be found in [32–37].

It was proved in [30] that, in the case of an integral domain K , the girth of $D(n, K)$ is bounded below by $n + 5$. Therefore, we define our algorithms in terms of $D(n, K)$ for finite commutative ring K .

2. On Protocols of Postquantum Cryptography and Subgroups of the Affine Cremona Group

Communication theory widely uses finite commutative rings $(K, +, \times)$ for the representation of information as tuples of kind $v = (p_1, p_2, \dots, p_n) \in K^n$, where coordinates p_i are elements of the alphabet K . Natural examples of commutative rings include the modular ring Z_{256} and the finite field F_{256} , whose order coincides with the size of the binary alphabet. The family of commutative rings of kind Z_p for prime p and Z_{pq} for two prime numbers p and q is popular because of their use in the Diffie–Hellman protocol and RSA algorithm.

Passwords in the form of tuples v are used in symmetric encryption and as access data to use the resources of an Information System. The key-establishment algorithm is a secure procedure which allows correspondents Alice and Bob to establish a common tuple of kind $v = (v_1, v_2, \dots, v_k)$ with coordinates from finite alphabet K .

The Diffie–Hellman algorithm can be used for key establishment. It employs a generator a of a cyclic subgroup of the multiplicative group Z_p^* , where p is a prime number and $(p - 1, a) = 1$. The protocol allows two parties to establish a shared secret b , which can be used for key establishment. Alice selects her private parameter $k(A)$. She computes $g(A) = a^{k(A)}$ and sends this element to Bob. He chooses his private key $k(B)$, computes $g(B) = a^{k(B)}$, and sends it via an open channel to Alice. At the next stage Alice computes the collision element b as $(g^{k(B)})^{k(A)}$ and Bob computes it as $(g^{k(A)})^{k(B)}$.

The piece of information b can be naturally converted into a tuple. The correspondents select $d, d < p$ and use the representation of $b = b_0 + b_1d + \dots + b_sd^s$, where b_i are elements of Z_d . This famous algorithm rests on the complexity of the discrete logarithm problem about the solution of equation $a^x = b$ in the multiplicative group of finite prime field F_p .

No polynomial-time solution to this problem is known. The security of the protocol rests on this fact. Assume that the fundamental conjecture of Complexity theory $P \neq NP$, i.e., class of problems solvable in polynomial time differing from the class of NP -hard problems, is true (see [18]).

Then the conjecture that the discrete logarithm problem is in the class of NP -hard problems is still under investigation. To justify this fact one has to prove the computational equivalence of DLP to one of the problems from well-known list of NP -hard problems. This has not yet been achieved. Other approaches to key establishment can be found in [38–43]. The theoretical possibility of solving the DLP problem in polynomial time on a quantum computer was discovered by Peter Shor. It makes the DH key-establishment algorithm insecure in our postquantum Era. Note that the Diffie–Hellman algorithm can be implemented with the generator from the semigroup S . This implementation heavily depends on the way of presenting S in the memory of the Turing machine. If we replace the group F_p^* by the isomorphic additive group Z_{p-1} , the discrete logarithm problem transforms to a simple Diophantine equation. One of the ideas of noncommutative cryptography is to develop postquantum secure protocols with the use of two or more noncommuting generators of the semigroup. The following scheme is a Twisted Diffie–Hellman protocol for the semigroup with unity S (monoid). *Input.* Alice and Bob share elements $g \in S$, an invertible element $h \in S$, and its inverse h^{-1} .

Step

1. (for Alice).

She chooses positive integers $k(A)$ and $r(A)$ and computes $h^{-r(A)}g^{k(A)}h^{r(A)} = g_A$.

Step 1. (for Bob).

Bob takes $k(B)$ and $r(B)$ and computes $h^{-r(B)}g^{k(B)}h^{r(B)} = g_B$.

Step 2.

They exchange g_A and g_B via an open channel.

Step 3. (for Alice).

She computes $X = h^{-r(A)}g_B^{k(A)}h^{r(A)}$.

Step 3. (for Bob).

He computes $Y = h^{-r(B)}g_A^{k(B)}h^{r(B)}$.

Output. The collision element $X = Y$ shared by Alice and Bob.

Remark 1. To break this scheme, an adversary has to solve the equation $h^y g^x h^{-y} = g_A$ or $h^y g^x h^{-y} = g_B$ for the unknowns x and y , i.e., to solve the conjugacy power problem (CPP) in the monoid M . Therefore, we seek monoids M for which no polynomial-time algorithm for solving the CPP is known.

Several candidates of groups for which CPP is unfeasible were rejected. We have to note that, similar to DLP, the complexity of the CPP depends on the representation of the monoid. In many investigated cases the monoid M is given by generators and relations. General concepts and constructive examples of algebraic cryptosystems are presented in [44–59]. Noncommutative Cryptography has its own history, it is the area of intensive new research. Corresponding cryptanalysis can be found in [60–66]. However, the search for appropriate platforms defined in terms of Combinatorial Group Theory continues.

We are looking for an alternative approach based on large subgroups or subsemigroups of the famous affine Cremona semigroup ${}^nCS(K)$ of endomorphisms of $K[x_1, x_2, \dots, x_n]$, where K is a finite commutative ring (see [67–69] and further references). It is a monoid acting on the affine space K^n as a semigroup of polynomial transformations. This object is important for the theory of symbolic computations [70].

An element σ of the semigroup can be specified by its values on the variables, i.e., as the rule $x_i \rightarrow f_i(x_1, x_2, \dots, x_n)$, $i = 1, 2, \dots, n$. This rule induces the map

$$\sigma' : (x_1, x_2, \dots, x_n) \rightarrow (f_1(x_1, x_2, \dots, x_n), f_2(x_1, x_2, \dots, x_n), \dots, f_n(x_1, x_2, \dots, x_n))$$

on the affine space $(K)^n$. We assume that σ is given in its standard form which is the list of f_i written as sums of monomial terms, ordered lexicographically. We define the *density* of f_i as the number of its monomial terms and the density $\text{den}(\sigma)$ of the endomorphism σ as maximum of densities of f_i .

We define *global density* $\text{gden}(\sigma)$ as the sum of densities of $f_i, i = 1, 2, \dots, n$. Degrees and densities of multivariate maps affect the complexity of the compositions of two elements. The situation regarding the complexity of the Conjugation Power Problem in the case of the affine Cremona semigroup is similar to the case of DLP for prime finite field F_p , i.e., the algorithm for solving the problem in polynomial time is unknown. The difference between these two cases is that, for $\text{CS}_n(K)$, no polynomial-time quantum algorithm for solving the problem is currently known. These facts justify constructions of protocols whose security rests on the following heuristic hypothesis.

CONJECTURE. The Conjugacy Power Problem for the Cremona semigroup $K[x_1, x_2, \dots, x_n]$ over a finite commutative ring with unity is *NP-hard*.

The implementation of the Diffie–Hellman protocol or its twisted analogue with generators in general position from ${}^n\text{CS}(K)$ is very costly because the product of two elements of degree r and s has degree rs . Recall that the degree of σ is the maximum of degrees of f_i . Consequently, the representation of the n -th power of a nonlinear element usually grows exponentially.

The discovery of the graphs $D(n, q)$ and their generalisations over commutative rings led to the construction of large subgroups of ${}^n\text{CS}(K)$ with the maximal degree of elements bounded by a constant. We refer to such subgroups as *stable subgroups* of the affine Cremona semigroup. The first example of such a subgroup was constructed by A. Wroblewska. The announcement of this result is given in [71]; the proof of the stability of the subgroup $G(n, q)$ of ${}^n\text{CS}(K)$ defined via walks on the graph $D(n, K)$ the reader may find in [72]. Note that, for the stable subgroup G of ${}^n\text{CS}(K)$, its conjugate $\sigma G \sigma^{-1}$ by an element σ from $\text{AGL}_n(K)$ is also a stable subgroup. Large stable subgroups of affine Cremona group are convenient platforms for the implementation of protocols of Noncommutative Cryptography. In most cases the theoretical complexity of the protocol coincides with the complexity of computing the product of two elements of degree d where $d = \text{deg}(G)$, i.e., the maximal degree of elements of G . This complexity can be evaluated as $O(n^{d^2+d+1})$ or more precisely as $O(n^{td+t+1})$, where the density of G is $O(n^t)$. It is noteworthy that $\text{den}(\sigma G \sigma^{-1})$ depends on the choice of σ . Estimates of these densities can be obtained by computer simulation.

The implementation of a protocol based on the platform of the form $\sigma G(n, K) \sigma^{-1} G(n, K)$ whose security rests on the Navigation Problem of decomposing an element $g \in G(n, K)$ is described in [73]. Its complexity is analysed for the rings $K = F_q, K = \mathbb{Z}_q, q = 2^{32}$ and the Boolean ring $B(32)$ of order 2^{32} . For each of these commutative rings, the cases in which the linear map σ is the identity, the density of σ is $O(n)$, and $\text{gden}(\sigma) = n^2$ are investigated. Estimates of the densities in the case where σ has density $O(n)$ are also given there. These estimates can be used for the Twisted Diffie–Hellman protocol whose complexity is also determined by the complexity of the composition of elements of $\sigma G(n, K) \sigma^{-1}$.

We present a subgroup $G(n, K)$ of cubic transformations of ${}^n\text{CS}(K)$ in Section 3. Other examples of stable subgroups and subsemigroups of ${}^n\text{CS}(K)$ are defined via walks on the Schubert orbitals of Tits geometry presented above (see [26]). One of the orbitals in the case of Coxeter–Dynkin diagram A_{2k} is the Double Schubert Graph $DS(k, F_q)$. It is a bipartite graph with partition sets isomorphic to $(F_q)^{n+n^2}$. A point $(x_1, x_2, \dots, x_k, x_{1,1}, x_{1,2}, \dots, x_{k,k})$ is incident to a line $[y_1, y_2, \dots, y_k, y_{1,1}, y_{1,2}, \dots, y_{k,k}]$ if and only if $x_{i,j} - y_{i,j} = x_i y_j$.

This graph was used for the construction of stable subgroups and subsemigroups of ${}^n\text{CS}(F_q)$ of prescribed degree d , where $d \geq 2$ [26]. The groups in the second case were used as a platform for the implementation of the Twisted Diffie–Hellman protocol [74].

The complexity of this protocol for users is $O(n^7)$. Instead of the graphs $D(n, q)$, their homomorphic images $((A(m, q), B(n, m, q))$ and some others [25]) can be used for the generation of other stable cubic subgroups of ${}^nCS(F_q)$.

In the next section we present a special version of the Twisted Diffie–Hellman protocol scheme (P_1) for the case of an abstract monoid S acting as a transformation semigroup on some set N . The advantage of this scheme in the case when $S < {}^nCS(K) >$ is the reduction in the number of costly multiplication operations in S . As in the general scheme of the Twisted Diffie–Hellman protocol presented above, these operations are used for the computation of g_A and g_B . However, instead of computing a collision element in S via multiple uses of costly operations, the participants, Alice and Bob, compute its value on an agreed element $m \in N$ without using the semigroup product. Therefore, the complexity of the implementation of the new scheme with the stable subgroup of the affine Cremona semigroup is still of the form $C(\text{den}(G)^d + 1)n$, but with a smaller constant C , due to the reduction in the number of multiplications of endomorphisms of $K[x_1, x_2, \dots, x_n]$ given in their standard forms. Protocol P_1 can be converted into the cryptosystem E_1 , which can be used as an encryption procedure. Other schemes (P_2, E_2) work with the abstract semigroup S and its homomorphism ϕ into ${}^nCS(K)$. If the cost of a single operation in S is less than the cost of the corresponding operation in $\phi(S)$, then these schemes have better theoretical estimates than P_1 and E_1 .

3. Algorithms for Transformation Semigroups

3.1. Scheme P_1 of the Twisted Diffie–Hellman Protocol for Transformation Semigroups

Let us assume that a monoid S acts on a finite set N , which is “sufficiently large”.

Input.

Alice and Bob share a randomly or pseudorandomly selected element $m \in N$, together with generators g, h , and h^{-1} from S .

Step 1. (for Alice). She selects positive integers $k(A)$ and $r(A)$ and computes g_A according to the scheme described in Section 2.

Step 1 (for Bob). He selects positive integers $k(B)$ and $r(B)$ and computes g_B according to the scheme described in Section 2.

Step 2. Alice and Bob exchange g_A and g_B via an open communication channel.

Step 3. (for Alice). She computes the following sequence consecutively:

$$h^{-1}(m) = {}^1m, h^{-1}({}^1m) = {}^2m, \dots, h^{-1}({}^{r(A)-1}m) = a,$$

$$g_B(a) = {}^1a, g_B({}^1a) = {}^2a, \dots, g_B({}^{k(A)-1}a) = a_1$$

and

$$h(a_1) = {}^1a_1, h({}^1a_1) = {}^2a_1, \dots, h({}^{r(A)-1}a_1) = a_2.$$

Step 3. (for Bob). He computes the following sequence analogously to Alice:

$$h^{-1}(m) = {}^1m, h^{-1}({}^1m) = {}^2m, \dots, h^{-1}({}^{r(B)-1}m) = b;$$

$$g_A(b) = {}^1b, g_A({}^1b) = {}^2b, \dots, g_A({}^{k(B)-1}b) = b_1$$

and

$$h(b_1) = {}^1b_1, h({}^1b_1) = {}^2b_1, \dots, h({}^{r(A)-1}b_1) = b_2.$$

Output. Alice and Bob obtain the collision element $c = a_2 = b_2$.

Remark 2. In fact, the participants can apply the constructed elements g_A and g_B to $O(1)$ pseudo-random elements $m_i, i = 1, 2, \dots, k$ from N and obtain several collision vectors. We use this scheme in the case when the monoid S is a stable subsemigroup of $CS_n(K)$ acting naturally on K^n .

3.2. Cryptosystem E_1

Let us assume that a monoid S acts on a finite set N .

Input.

Alice and Bob share invertible generators g, h , and h^{-1} from S . Alice also knows g^{-1} .

Step 1. (for Bob). He selects $r(B)$ and $k(B)$ and uses the procedure of *Step 1* of protocol P_1 .

Step 1. (for Alice). She selects $r(A)$ and $k(A)$ and computes $h^{-r(A)}g^{-k(A)}h^{r(A)} = g'_A$. Here, $g^{-k(A)}$ denotes the $k(A)$ -th power of g^{-1} .

Step 2. Alice and Bob exchange g'_A and g_B via an open communication channel.

Step 3. (for Bob). He forms a plaintext $m \in N$ and constructs the ciphertext $c(B) = h^{r(B)}((g'_A)^{k(B)}(h^{-r(B)}(m)))$ and sends it to Alice via an open channel.

Step 3. (for Alice) She receives the ciphertext $c(B)$ and decrypts it via the computation $m = h^{r(A)}((g'_B)^{k(A)}(h^{-r(A)}(c(B))))$.

Step 3'. (for Alice). She forms a plaintext $m \in N$ and constructs the corresponding ciphertext $h^{r(A)}((g_B)^{k(A)}(h^{-r(A)}(m))) = c(A)$.

Step 3'. (for Bob). He receives $c(A)$ and decrypts it via $h^{k(B)}(g'_A)^{k(B)}(h^{-k(B)}(c(A))) = m$.

Output is the following communication scheme: Bob uses *Step 3* to send his ciphertext to Alice, and she uses her *Step 3* to decrypt the message from Bob. Similarly, Alice uses her *Step 3'* to encrypt and send a message to Bob, and he uses his *Step 3'* to decrypt it.

Remark 3. We work with a stable subgroup S of the affine Cremona semigroup ${}^nCS(K)$ acting on the affine space K^n . Thus, K serves as the alphabet for writing plaintexts.

Remark 4. It is noteworthy that the execution of *STEP 3* and *STEP 3'* for Alice and Bob does not require the use of compositions of endomorphisms of $K[x_1, x_2, \dots, x_n]$.

3.3. Protocol 2 (P_2)

Let us assume that S is an abstract semigroup and that ϕ is a homomorphism of S into the Cremona semigroup ${}^nCS(K)$. Assume that H denotes the image $\phi(S)$, which is a monoid.

Input. Alice and Bob share elements g, h , and h' from S such that $\phi(hh')$ is the identity. They share a pseudorandom tuple $v \in K^n$.

Step 1. (for Alice) She selects her private numbers $r(A)$ and $k(A)$ and computes

$$s(A) = h^{r(A)}g^{k(A)}(h')^{r(A)}$$

together with its homomorphic image $g(A) = \phi(s(A))$.

Step

1. (for Bob) He selects his private numbers $r(B), k(B)$ and computes

$$s(B) = h^{r(B)}g^{k(B)}(h')^{r(B)}$$

together with its homomorphic image $g(B) = \phi(s(B))$.

Step 2. (for Alice) She computes $h(A) = \phi(h^{r(A)})$ and $h'(A) = \phi((h')^{r(A)})$ and keeps them in a secure way.

Step 2. (for Bob) He computes $h(B) = \phi(h^{r(B)})$ and $h'(B) = \phi((h')^{r(B)})$ for his private use only.

Step 3. (Alice and Bob) They exchange $g(A)$ and $g(B)$ via an open communication channel. They have the two following options

Step 4. (Alice) She computes the collision vector $c(v)$ via the following consecutive computations:

$$h(A)(v) = {}^1v, g(B)({}^1v) = {}^2v, h'(A)({}^2v) = {}^3v.$$

Step 4. (Bob) He computes

$$h(B)(v) = {}^1u, g(A)({}^1u) = {}^2u, h'(B)({}^2u) = {}^3u.$$

Output Alice and Bob privately keep $c(v) = {}^3v = {}^3u$.

Remark 5. They can select several values of v and keep several collision vectors $c_i, i = 1, 2, \dots, k$.

3.4. Protocol P_2'

Input.

Alice and Bob share elements g, h , and h' from S such that $\phi(hh')$ is the identity. Alice and Bob execute *Step 1*, *Step 2*, and *Step 3* of protocol P_2 . Instead of *Step 4*, they execute

Step 4'. (Alice) She computes the standard form of the composition $C(A)$ of $h(A)$, $g(B)^{k(A)}$, and $h'(A)$.

Step 4'. (Bob) He computes the $C(B) = h(B)(g(A))^{k(B)}h'(B)$ in its standard form.

Output. The collision map is $C = C(A) = C(B)$.

Remark 6. Assume that $\phi(S)$ is a stable monoid of ${}^nCS(K)$ of degree d . Then the number of nonzero coefficients in the standard form for C is its global density, evaluated as $O(n^{d+1})$. Thus, the participants can use the sequence of these coefficients ordered lexicographically as the output of P_2' . Alice and Bob may use $\sigma\phi(S)\sigma^{-1}$, where $\sigma \in AGL_n(K)$, instead of $\phi(C)$, and the array of nonzero coefficients of $\sigma C\sigma^{-1}$ as the output of the protocol.

Remark 7. The complexity of computation of value $g(B)$ or $g(A)$ on given tuple coincides with global densities of these maps which is $O(n^2)$ if $K = F_q$, $K = Z_q$ or $K = B(q)$, $q = 2^t$, according to the abovementioned computer experiment and our computations presented in the figures and tables of this paper. So the heuristic complexity of protocol in these cases is $O(n^2)$.

The general bound $O(n^4)$ for general finite rings follows from Proposition 2 of the previous section.

We also consider the following cryptosystems of El Gamal type, which are modifications of the schemes in [75].

3.5. Cryptosystem E_2

Assume that S is an abstract semigroup, that ϕ is a homomorphism of S into ${}^nCS(K)$, and that its image H is a monoid whose elements are mutually invertible.

Input.

Alice and Bob share g, h , and h' such that $\phi(h')$ is the inverse of $\phi(h)$ and $\phi(g)$ is an invertible endomorphism. Alice knows element g' such that $\phi(gg')$ is the identity.

Step 1. (Alice) She selects her private positive integers $r(A)$ and $k(A)$. She computes $h(A) = \phi(h^{r(A)})$ and $h'(A) = \phi((h')^{r(A)})$ and computes

$$g(A) = \phi(h^{r(A)})(g')^{k(A)}(h')^{r(A)}.$$

Step 1. (Bob) He acts as in the case of protocol P_2 , selecting $r(B)$ and $k(B)$ and computing

$$g(B) = \phi(h^{r(B)})g^{k(B)}(h')^{k(B)}.$$

Bob also computes $h(B) = \phi(h^{r(B)})$ and $h'(B) = \phi((h')^{r(B)})$.

Step 2. (Alice and Bob) They exchange $g(A)$ and $g(B)$ via an open communication channel. After the exchange of $g(A)$ and $g(B)$ correspondents, they have two options.

Step 3. (Alice) Alice writes her message p . She computes 1p as $h(A)(p)$, 2p as $g(B)^{k(A)}(^1p)$ (via consecutive application of $g(B)$ to the vector), and 3p as $h'(A)(^2p)$. Alice sends the ciphertext 3p to Bob.

Step 3. (Bob) He computes $^1c = h(B)(^3p)$. Then $^2c = g(A)^{k(B)}(^1c)$ (via the consecutive application of $g(A)$ to the vector) and $^3c = h'(B)(^2c)$. We can see that 3c coincides with the plaintext p of Alice.

Step 3'. (Bob) He writes his message p and computes 1p as $h(B)(p)$, 2p as $g(A)^{k(B)}(^1p)$ (via consecutive application of $g(A)$ to vector), and 3p as $h'(B)(^2p)$. He sends the ciphertext 3p to Alice.

Step 3'. (Alice) She computes $^1c = h(A)(^3p)$, $^2c = g(B)^{k(A)}(^1c)$ (via the consecutive application of $g(B)$ to the vector), and $^3c = h'(A)(^2c)$. We can see that 3c coincides with the plaintext p of Bob.

Output is the following communication scheme: Alice uses *Step 3* to send her ciphertext to Bob and he uses his *Step 3* to decrypt the message from Alice. Similarly, Bob uses his *Step 3'* to encrypt and send a message to Alice, and she uses her *Step 3'* to decrypt it.

Remark 8. The security of Protocol P_2 and Cryptosystem E_2 is based on the complexity of the Conjugacy Power Problem for elements $\phi(h)$ and $\phi(g)$ of $\phi(S)$.

We assume that the cost of a single operation in S is lower than the cost of an operation in $\phi(S)$. In the case of a stable $\phi(S)$ of degree d , this cost is evaluated in Section 2. The following obfuscations of the protocols and cryptosystems presented in this section are as follows.

Remark 9 (Desynchronised mode). Alice and Bob share distinct elements 1L and 2L from the general affine group $AGL_n(K)$. They can use $^1L\phi(h)^1L^{-1}$ and $^2L\phi(g)^2L^{-1}$ instead of the generators $\phi(h)$ and $\phi(g)$ in the semigroup ${}^nCS(K)$.

Remark 10 (Hidden semigroup S). Alice uses her knowledge of S and ϕ , but she sends the elements $^1L\phi(h)^1L^{-1}$ and $^2L\phi(g)^2L^{-1}$ in their standard form to Bob. Thus, he uses the Twisted Diffie–Hellman protocol and its inverse version with these generators in their standard forms.

4. Walks on Infinite Graphs and Cubic Transformations of Affine Spaces

Note that quadratic Equations (1) defining the infinite graph $D(K)$ provide a description of the neighbourhood of each vertex via an infinite linear system written in row echelon form. If we take the point $(c) = (c_1, c_2, \dots)$, then its neighbourhood is described by Equations (1) with variables p_i specialised by the conditions $p_i = c_i$, while the variables l_i remain free. Thus, choosing $l_1 = a$, where $a \in K$, determines the neighbour $N_a((c))$ as the unique solution of this system of linear equations. We refer to the first coordinate c_1 of a point (c) (or of a line $[c] = [c_1, c_2, \dots]$) as the colour of the point (or line, respectively).

Thus, we obtain a well-defined operator $N_a(v)$ that assigns to v its neighbour of colour a . Let $I = I(K)$ denote the incidence relation of the bipartite graph $D(K)$. Given a fixed point (u) of colour b and a sequence

$$s = (a_1, b_1, a_2, b_2, \dots, a_k, b_k)$$

of elements of K , we obtain a walk in the graph of the form

$$uIv_1Iu_1Iv_2Iu_2I\dots v_kIu_k$$

, where each v_i is a point of colour b_i and each u_i is a line of colour a_i .

The standard homomorphism $\eta = \eta_n$ from $D(K)$ onto $D(n, K)$, obtained by deleting all coordinates with indices greater than n , allows us to consider the sequence

$$\eta(u), \eta(v_1), \eta(u_1), \eta(v_2), \eta(u_2), \dots, \eta(v_k), \eta(u_k).$$

The graph $D(K)$ is defined over an arbitrary commutative ring K . Therefore, we may consider the graph $\widetilde{D(K)} = D(K[x_1, x_2, \dots, x_n, \dots])$, where $x_i, i = 1, 2, \dots$ is an infinite set of indeterminates.

If K is an integral domain, i.e., a commutative ring without zero divisors, then both graphs $D(K)$ and $\widetilde{D(K)}$ are forests. Even when K is finite, the neighbourhood of each vertex of $\widetilde{D(K)}$ is infinite.

We consider an induced subgraph D' of $\widetilde{D(K)}$ consisting of all vertices whose colours belong to $K[x_1]$. We take some point of D' of kind

$$(p) = (f(x_1), f_2(x_1, x_2, \dots), f_3(x_1, x_2, \dots), \dots)$$

to be a point of colour $f(x_1)$, and we let w be a sequence of elements

$$(A_1(x_1), B_1(x_1), A_2(x_1), B_2(x_1), \dots, A_k(x_1), B_k(x_1))$$

of $K[x_1]$. This sequence defines a walk, also denoted by w , in D' with initial point p , whose subsequent vertex colours are given by

$$B_1(f(x_1)), A_1(f(x_1)), B_2(f(x_1)), A_2(f(x_1)), \dots$$

We refer to the sequence w as a symbolic word. If w' is another symbolic word

$$w' = (A'_1(x_1), B'_1(x_1), A'_2(x_1), B'_2(x_1), \dots, A'_l(x_1))$$

, then the walk w can be extended from its terminal vertex, denoted by 1p , of colour $B_k(f(x_1))$ (by consecutively adding neighbours whose colours are

$$A'_1(B_k(f_1(x_1))), B'_1(B_k(f_1(x_1))), A'_2(B_k(f_1(x_1))), B'_2(B_k(f_1(x_1))), \dots$$

This construction motivates the following definition of the composition of the symbolic words w and w' :

$$(A_1(x_1), B_1(x_1), A_2(x_1), B_2(x_1) \dots A_k(x_1), B_k(x_1), \\ A'_1(B_k(x_1)), B'_1(B_k(x_1)), A'_2(B_k(x_1)), B'_2(B_k(x_1)), \dots, A'_l(B_k(x_1)), B'_l(B_k(x_1)))$$

This operation defines the semigroup $SD(K)$. We observe that this semigroup is the semidirect product of the free semigroup generated by the alphabet $K[x_1]$ and the Cremona semigroup ${}^1CS(K)$. We define a homomorphism

$$\phi : SD(K) \rightarrow \text{End}(K[x_1, x_2, \dots])$$

such that, for

$$w = (A_1, B_1, A_2, B_2, \dots, A_k, B_k)$$

the image $\phi(w)$ is the endomorphism that sends the tuple $(x_1, x_2, x_3, \dots)$ to $(u_1, u_2, u_3, \dots)$, where $(u_1, u_2, u_3, \dots)$ is the terminal point of the walk of length $2k$ starting at $(x) = (x_1, x_2, \dots)$ and passing through vertices whose colours are $A_1, B_1, A_2, B_2, \dots, A_k, B_k$ of the graph $D'(K)$. The map $\eta_n(\phi)$ defines a homomorphism ϕ_n from $SD(K)$ onto ${}^nCS(K)$. The endomorphisms $\phi(w)$ and $\phi_n(w)$ are automorphisms if and only if B_k is an automorphism of $K[x_1]$. (see [26] and the references therein).

Assume that $W(K) = \phi(CD(K))$ and $W(n, K) = \phi_n(CD(K))$.

For each symbolic word

$$w = (A_1, B_1, A_2, B_2, \dots, A_k, B_k)$$

we define w^* to be the tuple

$$(B_k^{-1}(A_k), B_k^{-1}(B_{k-1}), B_k^{-1}(A_{k-1}), \dots, B_k^{-1}(B_1), B_k^{-1}(A_1), B_k^{-1}).$$

One can verify that $\phi(w \times w^*) = e$ and $\phi_n(w \times w^*) = e$ for all $n \geq 2$.

Let us consider the subsemigroup $S(K)$ of $CD(K)$ consisting of symbolic words whose coordinates are of the form $x_1 + c$, $c \in K$. Then $G(K) = \phi(S(K))$ and $G(n, K) = \phi_n(S(K))$ are subgroups of $W(K)$ and $W(n, K)$, respectively. The following result was announced in [71] and proved in [72].

Theorem 1. *Every element of $G(n, K)$ has at most a degree of 3.*

We encode symbolic words of the form

$w = (x_1 + a_1, x_1 + b_1, \dots, x_1 + a_k, x_1 + b_k)$ for which

$w^* = (x_1 - b_k + a_k, x_1 - b_k + b_{k-1}, x_1 - b_k + a_{k-1}, \dots, x_1 - b_k + a_1, x_1 - b_k)$

and represent them by

$$(a_1, a_2, \dots, a_k; b_1, b_2, \dots, b_k)$$

The multiplication rule for symbolic words in this notation is given by the following formula.

$$\begin{aligned} (a_1, a_2, \dots, a_k; b_1, b_2, \dots, b_k) \times (a'_1, a'_2, \dots, a'_l; b'_1, b'_2, \dots, b'_l) = \\ (a_1, a_2, \dots, a_k, a'_1 + b_k, a'_2 + b_k, \dots, a'_l + b_k; b_1, b_2, \dots, b_k, b'_1 + b_k, b'_2 + b_k, \dots, \\ b'_l + b_k) \end{aligned} \quad (2)$$

We refer to w , written in the new notation, as an *irreducible symbolic word* if $a_i \neq a_{i+1}$ and $b_i \neq b_{i+1}$ for $i = 1, 2, \dots, k-1$ and define the length $l(w)$ of an irreducible symbolic word to be $2k$.

The expression for w^* in the new notation is immediate.

It can be shown that the global density $gden(\phi_n(w)) = gden(n, l)$ depends only on the length $l(w)$ and that

$$den(\phi_n(w)) = \frac{gden(n, l)}{n}.$$

In [73], the quantities $gden(n, k)$ were investigated for $K = F_q$, $K = Z_q$ and for the Boolean ring $K = B_q$ of order q , where $q = 2^r$ and $r \leq 32$. Let $gden(n, l, K)$ denote the quantity $gden(n, l)$ computed over the commutative ring K . The following proposition is an immediate corollary of Theorem 1.

Proposition 1. $gden(n, k, K) = O(n^4)$.

The following properties were supported by computer simulations for $K \in \{F_q, Z_q, B_q\}$, where q is a power of 2 (see [73]).

- (1) $gden(n, k, K) = O(n^2)$
- (2) $gden(n, k, K) = c(n)$ if $k \geq k(n)$ for some parameter $k(n)$.

It is shown there that Property (2) holds also for $gden(L\phi_n(w)L^{-1}) = glob(L, n, l, K)$, where w is a symbolic word over K of length l and L is a special linear transformation over K . The following two cases were selected.

- (a) $L(x_1) = x_1 + a_2x_2 + a_3x_3 + \dots + a_nx_n$, $L(x_i) = x_i$ for $i \geq 2$
- (b) L is represented by a nonsingular matrix whose entries are all nonzero.

It turns out that

$$gden(L, n, l, K) = O(n^3)$$

in case (a), whereas

$$gden(L, n, l, K) = O(n^4)$$

in case (b).

5. Protocols and Cryptosystems Based on the Platform $\phi_n(S(K))$

For the implementation of Protocol P_2 and Cryptosystem E_2 , we use the semigroup $S(K)$ and the homomorphism ϕ_n , where K is a commutative ring with unity. We assume that the parameters $k(A)$ and $k(B)$ are significantly larger than the characteristic of the ring.

5.1. Protocol P_2

Input.

Alice and Bob share a finite commutative ring K , the graph $D(K)$, and a positive integer n . They also share two pseudorandom sequences $w(1) = (a_1, a_2, \dots, a_k, b_1, b_2, \dots, b_k)$ and $w(2) = (a'_1, a'_2, \dots, a'_l, b'_1, b'_2, \dots, b'_l)$ consisting of elements of K such that $a_i \neq a_{i+1}, b_i \neq b_{i+1}$ for $i = 1, 2, \dots, k-1$ and $a'_i \neq a'_{i+1}, b'_i \neq b'_{i+1}$ for $i = 1, 2, \dots, l-1$. These sequences will be used as symbolic words in the semigroup $S(K)$ and are both known to be correspondents. In the case of a ring of large characteristic, the symbolic word $w(1)$ should be chosen so that $b_k = 0$. Additionally, they share a pseudorandom element $p = (p_1, p_2, \dots, p_n) \in K^n$.

Step 1. (Alice) She computes the symbolic word $w(2)^*$ using the explicit rule given in the previous section. Alice selects positive integers $r(A)$ and $k(A)$ and uses the composition rule from Proposition 1 to compute $(w(2))^{r(A)} = w(A)$, $(w(2)^*)^{r(A)} = w'(A)$ and symbolic word $w(A)w(1)^{k(A)}w'(A) = u(A)$.

Step 1. (Bob) He computes words $w(2)^*$ in $S(K)$. Bob selects his private integers $r(B)$ and $k(B)$ and computes the symbolic words $(w(2))^{r(B)} = w(B)$, $(w(2)^*)^{r(B)} = w'(B)$ and $w(B)w(1)^{k(B)}w'(B) = u(B)$.

Step 2. (Alice) She computes $\phi_n(w(A)) = h(A)$, $\phi(w'(A)) = h'(A)$ and $g(A) = \phi(u(A))$. Recall that, in order to compute $\phi(c_1, c_2, \dots, c_s, d_1, d_2, \dots, d_s)$, Alice considers the

graph $D(n, K[x_1, x_2, \dots, x_n])$. She takes the point $x = (x_1, x_2, \dots, x_n)$ and computes the walk in this graph of kind

$$\begin{aligned} N_{x_1+c_1}(x) &= v_1, N_{x_1+d_1}(v_1) = u_1, \\ N_{x_1+c_2}(u_1) &= v_2, N_{x_1+d_2}(v_2) = u_2, \\ &\dots, \\ N_{x_1+c_s}(u_{s-1}) &= v_s, N_{x_1+d_s}(v_s) = u_s. \end{aligned}$$

The terminal vertex u_s is the tuple $(f_1, f_2, \dots, f_n)$, $f_i \in K[x_1, x_2, \dots, x_n]$. The image under ϕ is the map $x_i \rightarrow f_i$, $i = 1, 2, \dots, n$ given in its standard form.

Step 2. (Bob). He computes $\phi_n(w(B)) = h(B)$, $\phi(w'(B)) = h'(B)$ and $g(B) = \phi(u(B))$.

Step 3. Alice and Bob. They exchange the maps $g(A)$ and $g(B)$.

Step 4. (Alice) She computes $w(A)(p) = {}^1p$, and she applies $g(B)$ repeatedly to the intermediate result to get $g(B)^{k(A)}({}^1p) = {}^2p$. Finally, Alice computes ${}^3p = w'(A)({}^2p)$.

Step 4. (Bob) He computes $w(B)(p) = {}^1r$, $g(A)^{k(B)}({}^1r) = {}^2r$. Finally, Bob computes ${}^3r = w'(B)({}^2r)$.

Output. The shared collision vector ${}^3p = {}^3r$.

Remark 11. The complexity of computing the value of ϕ_n on a given symbolic word of length $O(1)$ is $O(n)$. The complexity of evaluating $g(B)$ or $g(A)$ on the given tuple coincides with global densities of these maps which is $O(n^2)$ if $K = F_q$, $K = Z_q$ or $K = B(q)$, $q = 2^t$. So the complexity of the protocol in these cases is $O(n^2)$.

The general bound $O(n^4)$ for finite rings follows from Proposition 1 of the previous section.

5.2. CRYPTOSYSTEM E_2

Input.

Alice and Bob share a finite commutative ring K , the graph $D(K)$, and a positive integer n . They share two pseudorandom sequences $w(1) = (a_1, a_2, \dots, a_k, b_1, b_2, \dots, b_k)$ and $w(2) = (a'_1, a'_2, \dots, a'_l, b'_1, b'_2, \dots, b'_l)$ consisting of elements of K such that $a_i \neq a_{i+1}$, $b_i \neq b_{i+1}$ for $i = 1, 2, \dots, k-1$ and $a'_i \neq a'_{i+1}$, $b'_i \neq b'_{i+1}$ for $i = 1, 2, \dots, l-1$. Alice and Bob execute Steps 1, 2, and 3 of Protocol P_2 .

Step 4. (Alice). She selects a plaintext $p \in K^n$. She computes $w(A)(p) = {}^1p$, and she applies $g(B)$ repeatedly to the intermediate result to get $g(B)^{k(A)}({}^1p) = {}^2p$. Finally Alice computes the ciphertext ${}^3p = w'(A)({}^2p)$.

Step 4. (Bob). He gets the ciphertext 3p from Alice. Bob computes $w(B)({}^3p) = r$. He applies $g(A)$ iteratively to r to get $g(A)^{k(B)}(r) = {}^1r$. Finally, he computes $w'(B)({}^1r)$ which coincides with the plaintext p of Alice.

Step 4'. (Bob). He selects a plaintext $p \in K^n$. Bob computes $w(B)(p) = {}^1p$, and he consecutively applies $g(A)$ to the intermediate result to get $g(A)^{k(B)}({}^1p) = {}^2p$. Finally he computes the ciphertext ${}^3p = w'(B)({}^2p)$.

Step 4'. (Alice) She receives the ciphertext 3p from Bob. Alice computes $w(A)({}^3p) = r$. She applies $g(B)$ iteratively to r to get $g(B)^{k(A)}(r) = {}^1r$. Finally she computes $w'(A)({}^1r)$ which coincides with the plaintext p written by Bob.

Output. The resulting communication scheme is as follows. Alice uses Step 4 to send her ciphertext to Bob, who then uses Step 4 to decrypt the message. Similarly, Bob uses Step 4' to encrypt and send a message to Alice, and she uses his Step 4' to decrypt.

5.3. Protocol P'_2

The Input is the same as for Cryptosystem E_2 .

Alice and Bob execute Step 1, Step 2, and Step 3 of protocol P_2 . Instead of Step 4, they execute the following steps.

Step 4'(a). (Alice) She computes the composition

$$X = w'(A) \circ g(B)^{k(A)} \circ w(A)$$

in the affine Cremona semigroup.

Step 4'(b). (Bob) He computes the composition

$$Y = w'(B) \circ g(A)^{k(B)} \circ w(B)$$

in the affine Cremona semigroup.

Output. The collision map $X = Y = C$ shared by Alice and Bob. For $K \in \{\mathbb{F}_q, \mathbb{Z}_q, B(q)\}$, they may use the ordered list of the $\mathcal{O}(n^2)$ nonzero coefficients of the map as the shared output.

Remark 12. They may additionally share a special bijective linear transformation

$$L : K^n \rightarrow K^n.$$

and use the ordered nonzero coefficients of LCL^{-1} . In Section 2, we observed that we can get $\mathcal{O}(n^3)$ or $\mathcal{O}(n^4)$ nonzero coefficients by an appropriate choice of L .

Remark 13. The computational complexity of Protocol P'_2 for Alice and Bob is $(k(A) + 1)$ and $(k(B) + 1)$ times, respectively, the cost of a single operation of the semigroup $S(K)$ which is $\mathcal{O}(n^5)$ in the case $K \in \{\mathbb{F}_q, \mathbb{Z}_q, B(q)\}$.

Remark 14. In the simplest version of the hidden graph cryptosystem proposed in the previous section, Alice selects $w(1)$ and $w(2)$ and sends Bob the standard forms of the maps $g = L\phi(w(1))L^{-1}$ and $h = L\phi(w(2))L^{-1}$, where L is chosen from $AGL(n, K)$ such that the densities of L and L^{-1} are both $\mathcal{O}(1)$. Thus, they execute a twisted Diffie–Hellman protocol with generators g and h from $G(n, K)$. Even in the third case, the complexity of the protocol is at least $\mathcal{O}(n^5)$. We implemented this procedure for $K = \mathbb{F}_q$ of even characteristics with linear L given by the rule $L(x_1) = a_1x_1 + a_2x_2 + \cdots + a_nx_n$, $a_1 \in K^*$, $a_i \neq 0$, $L(x_i) = x_i$, $i \geq 2$. The complexity is $\mathcal{O}(n^9)$.

Remark 15. There is an alternative way to increase the size of the output. Alice and Bob agree on a linear transformation L via an open channel. Then they compute the transformation $CLC = Z$. If the matrix of L is in general position, then the cost of computing Z which has degree 9 is $\mathcal{O}(n^7)$ and its global density is cn^7 for some constant c . They share this large tuple of ordered nonzero coefficients of Z .

They can execute this procedure with various matrices generated with use of AI or Quantum Computing. If they use this procedure with a sparse transformation L of density $\mathcal{O}(1)$, then it costs $\mathcal{O}(n^5)$ elementary operations and the list of nonzero coefficients has length cn^5 .

Let us consider several cryptanalytic aspects for finite fields $\mathbb{F}_q$ for the implementation of protocol P_2 and cryptosystem E_2 in the case when the adversary knows the graph $D(n, K)$. In this case, the graph $D(\mathbb{F}_q)$ is an infinite regular forest and the graph $D(\mathbb{F}_q[x_1, x_2, \dots, x_n])$ is an infinite forest of infinite degree. The adversary is able to perform computations in the open semigroup $S(\mathbb{F}_q)$. Recall that two elements, $w(1)$ and $w(2)$, of even length of size $\mathcal{O}(1)$

of the semigroup $S(F_q)$ of random or pseudorandom nature are publicly known, together with the positive integer n . Thus, the adversary computes elements $h = \phi_n(w(1))$ and $g = \phi_n(w(2))$ in their standard forms. The adversary must solve the Conjugacy Power Problem with these generators and right-hand side $g(A)$ (or $g(B)$) of $G(n, q)$ for which no polynomial-time solution is known for the case of computations in ${}^nCS(F_q)$. Notice that the degrees of $g(A)$ and $g(B)$ are equal to 3.

Computer experiments in the investigated case of even q justify that the densities of $g(A)$ and $g(B)$ are $O(n)$ and for a sufficiently large $2r(A) + k(A)$ ($2r(B) + k(B)$, respectively) the densities of these maps are constant. Therefore, the adversary cannot use degrees or densities to estimate the sizes of the private keys. Alternatively, the adversary may use a pseudorandom message v (in the case of algorithm E the tuple $v = p$ is the plaintext) and try to look at walks between v and $g(A)(v)$ (or $g(B)$) to obtain some info on the possible size of $r(A)$ and $k(A)$ (or $r(B)$ and $k(B)$).

This option turns out to be computationally difficult because the complexity of finding the shortest path in the graph of order m is $m \log(m)$ (Dijkstra algorithm or other options). In our case, $m = 2q^n$ and complexity is exponential. Thus, this approach is computationally infeasible. In fact, for expanding Cayley graphs, some faster algorithms are known but graph $D(n, q)$ is not a Cayley graph. Below (Tables 1, 2 and Figure 3) we present the results of our computer simulations for the global density $gden(n, l, F_q^{24})$ for evaluating the global density of elements $\phi_n(w)$ corresponding to symbolic words of length $l, l \leq n$ which agrees with the results of [73]; our performance evaluation of the protocol was carried out on a MacBook Retina equipped with a 1.2 GHz Intel Core m3 processor and 8 GB of RAM.

Table 1. Total number of monomial terms of $\phi_n(w)$.

n	Length l			
	32	64	124	248
16	112	47	145	112
32	456	288	545	462
64	1815	1456	2113	1882
128	5399	6479	8321	7602
256	12,567	20,047	32,988	30,286

Table 2. Generation time for the map (ms).

n	Length l			
	32	64	124	248
16	12.71	20.41	39.85	70
32	35.6	62.8	132.4	252.9
64	97	234.5	567.2	1211
128	260.9	1121.9	3692.7	7300.8
256	640.2	3268.6	12,556.5	34,079.7

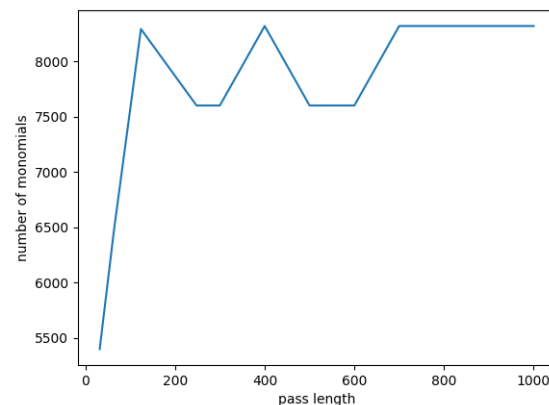


Figure 3. Number of monomial terms of the cubic map induced by the graph, $n = 128, F_{2^{24}}$.

6. Conclusions

Paper [24], published 30 years ago, presents the infinite q -regular tree as a well-defined projective limit $D(q)$ of the incidence structures $D(n, q)$, $n = 2, 3, \dots$ whose sets of points and lines are isomorphic to $(F_q)^n$ and the incidence relation given by a system of quadratic equations in terms of the coordinates of elements. The projective limit of the connected components $CD(n, q)$ is the q -regular tree.

We mentioned applications of these results to Extremal Graph Theory [76], the Theory of Algebraic Graphs [77], expanding graphs (see [78] and the references therein), the theory of LDPC codes, and cryptography. A recent paper [25] presents results on the spectra of $CD(n, q)$. We present new applications of these graphs to Noncommutative Cryptography aimed at the construction of a postquantum analogue of the Diffie–Hellman key-exchange protocol in terms of algebraic objects equipped with noncommutative binary operations. We use the semigroup $S(F_q)$ of even-length walks on $D(q)$ and its homomorphism onto the noncommutative group $G(n, q)$ of cubic transformations of the vector space F_q^n , which corresponds to one part of the bipartition of $D(n, q)$.

We propose a family of ElGamal-type protocols and algorithms whose security relies on the complexity of the Conjugacy Power Problem in the affine Cremona group of automorphisms of $F_q[x_1, x_2, \dots, x_n]$. The outputs of the protocols are given either as vectors from $(F_q)^n$ or as cubic transformations of this vector space with prescribed density ($O(n)$ or $O(n^2)$).

These forms of output allow the algorithms to be used as key-establishment procedures. The output vectors can be used as passwords for accessing an information system or as keys for a symmetric cipher.

For these algorithms, the computational complexity for each user is provided. In the fastest case, the complexity is $O(n^2)$ (the open-graph setting for which we present computer-simulation results). In the case of a protocol or cryptosystem with a hidden graph and a platform of the form $LG(n, K)L^{-1}$, where the densities of L and L^{-1} are both $O(1)$, the execution of the algorithm requires $O(n^5)$ operations in the underlying commutative ring.

In the case when $L(x_1)$ is a general linear combination of variables and $L(x_i) = x_i$ for $i \geq 2$, the complexity of the algorithms is estimated as $O(n^9)$. It compares well with the complexity $O(n^{13})$ of protocols based on the Navigation Problem with the same platform of a group conjugated with $G(n, q)$ via a linear transformation (see [73]), but it is slower than the implementation [74] of the Twisted Diffie–Hellman algorithm with a platform of stable quadratic transformations, which has complexity $O(n^7)$. The protocols presented in this paper are defined in the general case of graphs $D(n, K)$, where K is an arbitrary finite

commutative ring. There is a possibility of replacing the graph $D(n, K)$ with other graphs over K defined in [25] via quadratic equations.

Such a change preserves the complexity estimates. We are not aware of research on applications of the graphs $D(n, K)$ to the construction of public keys. Recent results in this direction can be found in [7]. The search for efficient key-establishment algorithms that can serve as a reliable standard is an active area that is currently developing rapidly. The algorithms in [79–100] use alternative approaches to Noncommutative Cryptography. It is noteworthy that the security of lattice-based algorithms relies on a known NP-hard problem.

Author Contributions: Conceptualization, V.U. and T.C.; Methodology, V.U. and T.C.; Software, T.C.; Validation, T.C.; Formal analysis, V.U.; Investigation, V.U.; Resources, T.C.; Data curation, T.C.; Writing—original draft, V.U.; Writing—review and editing, V.U. and T.C.; Visualization, T.C.; Supervision, V.U. and T.C.; Project administration, V.U.; Funding acquisition, V.U. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the NCN, grant number: 2025/09/X/ST1/00790.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Brouwer, A.E.; Cohen, A.M.; Neumaier, A. *Distance-Regular Graphs*; Springer: Berlin/Heidelberg, Germany, 1989.
2. Betten, A.; Braun, M.; Kerber, A.; Kohnert, A.; Wasserman, A. *Error Correcting Linear Codes Isometry and Applications*; Springer: Berlin/Heidelberg, Germany, 2006.
3. Essenhans, A.S.; Kohnert, A.; Wassermann, A. Constructions of codes for Network Coding. *arXiv* **2010**, arXiv:1005.2839.
4. Bannai, E.; Bannai, E.; Ito, T.; Tanaka, R. *Algebraic Combinatorics*; Walter de Gruyter GmbH & Co KG: Berlin, Germany, 2021; p. 22.
5. Delsarte, P. *An Algebraic Approach to the Association Schemes of Coding Theory*, Philips Research Reports Supplements, No. 10; N.V. Philips' Gloeilampenfabrieken: Eindhoven, The Netherlands, 1973; pp. vi+97.
6. Beultespacher, A. Enciphered Geometry, Some Applications of Geometry to Cryptography. *Ann. Discret. Math.* **1988**, *37*, 59–68.
7. Ustimenko, V. On Schubert cells of Lie geometries and public keys of multivariate cryptography. *Contemp. Math. AMS* **2025**, *830*. <https://doi.org/10.1090/conm/830/16568>.
8. Buekenhout, F. (Eds.) *Handbook on Incidence Geometry*; North Holland: Amsterdam, The Netherlands, 1995.
9. Bourbaki, N. *Lie Groups and Lie Algebras*; Springer: Berlin/Heidelberg, Germany; Chapters 1–9, pp. 1998–2008.
10. Tits, J. *Buildings of Spherical Type and Finite BN-Pairs*; Lecture Notes in Math; Springer: Berlin/Heidelberg, Germany, 1974.
11. Carter, R.W. *Simple Groups of Lie Type*; Wiley: New York, NY, USA, 1972.
12. Moody, R.V. Lie algebras associated with generalized cartan matrices. *Bull. Am. Math. Soc.* **1967**, *73*, 217–222. <https://doi.org/10.1090/S0002-9904-1967-11688-4>.
13. Kac, V. *An Infinite Dimensional Lie Algebras*; Birkhäuser: Boston, MA, USA, 1983. <https://doi.org/10.1007/978-1-4757-1382-4>.
14. Gelfand, I.; Serganova, V. Combinatorial geometries and torus strata on homogeneous compact manifolds. *Russ. Math. Surv.* **1987**, *42*, 133–168.
15. Borovik, A.; Gelfand, I.; White, N. Combinatorial flag varieties *J. Comb. Theory (A)* **2000**, *91*, 111–136.
16. Ustimenko, V. On some properties of Chevalley groups and their generalisations. In *Investigations in Algebraic Theory of Combinatorial Objects*; Institute of System Studies: Moscow, Russia, 1985; pp. 134–138. (In Russian)
17. Ustimenko, V. On the Embeddings of Some Geometries and Flag Systems in Lie Algebras and Superalgebras. *Root Syst. Represent. Geom.* **1990**, *3*–16.
18. Garey, M.R.; Johnson, D.S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*; Computers; W. H. Freeman: New York, NY, USA, 1979; 338p.
19. Chevalley, C. Sur certains groupes simples. *Tohoku Math. J. Second Ser.* **1955**, *7*, 14–66.
20. Ustimenko, V. Linear interpretation of Chevalley group flag geometries. *Ukr. Math. J.* **1991**, *43*, 1055–1060.
21. Ustimenko, V. Affine System of Roots and Tits Geometries. *Voprosy Teorii Grupp i Gomologicheskoy Algebrы Yaroslavl* **1989**, 155–157.
22. Lazebnik, F.; Ustimenko, V. Explicit construction of graphs with arbitrary large girth and of large size. *Discret. Appl. Math.* **1995**, *60*, 275–284.

23. Lazebnik, F.; Ustimenko, V.; Woldar, A. A characterization of the components of the graphs $D(k, q)$. *Discret. Math.* **1996**, *157*, 271–283.
24. Lazebnik, F.; Ustimenko, V.; Woldar, A.J. A new series of dense graphs of high girth. *Bull. AMS* **1995**, *32*, 73–79.
25. Chojecki, T.; Erskine, G.; Tuite, J.; Ustimenko, V. On affine forestry over integral domains and families of deep Jordan-Gauss graphs. *Eur. J. Math.* **2025**, *11*, 10. <https://doi.org/10.1007/s40879-024-00798-2>.
26. Ustimenko, V. *Graphs in Terms of Algebraic Geometry, Symbolic Computations and Secure Communications in Post-Quantum World*; UMCS Editorial House: Lublin, Poland, 2022; 196p. Available online: <https://bazawiedzy.umcs.pl/info/book/UMCSb86978566ae5417e8eb5b25976d5fb2f/Publikacja%E2%80%93Graphs+in+terms+of+Algebraic+Geometry+symbolic+computations+and+secure+communications+in+Post-Quantum+world+%E2%80%93Uniwersytet+Marii+Curie-Sk%C5%82odowskiej?r=publication&ps=20&lang=en&pn=1&cid=633423> (accessed on 8 September 2025).
27. Guinand, P.; Lodge, J. Tanner Type Codes Arising from Large Girth Graphs. In Proceedings of the 1997 Canadian Workshop on Information Theory (CWIT '97), Toronto, ON, Canada, 3–6 June 1997; pp. 5–7.
28. Guinand, P.; Lodge, J. Graph theoretic construction of generalized product codes. In Proceedings of the IEEE International Symposium on Information Theory ISIT'97, Ulm, Germany, 29 June–4 July 1997.
29. MacKay, D.; Postol, M. Weakness of Margulis and Ramanujan Margulis Low Density Parity Check Codes. *Electron. Notes Theor. Comput. Sci.* **2003**, *74*, 8.
30. Ustimenko, V. On linguistic dynamical systems, families of graphs of large girth, and cryptography. *J. Math. Sci.* **2007**, *140*, 461–471. <https://doi.org/10.1007/s10958-007-0453-2>.
31. Polak, M.; Zhupa, E. Keyed hash function from large girth expander graphs. *Albanian J. Math.* **2022**, *16*, 25–39.
32. Paszkiewicz, A.; Kotulski, Z.; Kulesza, K.; Szczepanski, J. *Proposals of Graph Based Ciphers Theory and Implementations*; Research Gate: Berlin, Germany, 2001.
33. Costache, A.; Feigon, B.; Lauter, K.; Massierer, M.; Puskas, A. Ramanujan Graphs in Cryptography. In *Research Directions in Number Theory*; Balakrishnan, J., Folsom, A., Lalin, M., Manes, M., Eds.; Association for Women in Mathematics Series; Springer: Berlin/Heidelberg, Germany, 2026; Volume 19.
34. Priyadarsini, P.L.K. A Survey on some Applications of Graph Theory in Cryptography. *J. Discret. Math. Sci. Cryptogr.* **2015**, *18*, 209–217. <https://doi.org/10.1080/09720529.2013.878819>.
35. Etaïwi, W.M.A. Encryption Algorithm Using Graph Theory. *J. Sci. Rep.* **2014**, *3*, 2519–2527.
36. Charles, X.D.; Goren, E.Z.; Lauter, K.E. Cryptographic hash functions from expander graphs. *J. Cryptol.* **2009**, *22*, 93–113.
37. Tomkins, H.; Nevins, M.; Salmasian, H. New Zemor-Tillich type hash functions over $GL_2(F)$. *J. Math. Cryptol.* **2020**, *32*, 236–253.
38. Van Oorschot, P.C. Authentication Protocols and Key Establishment. In *Computer Security and the Internet*; Information Security and Cryptography; Springer: Cham, Switzerland, 2020. https://doi.org/10.1007/978-3-030-33649-3_4.
39. Abdalla, M. Password-Based Authenticated Key Exchange: An Overview. In *Provable Security. ProvSec 2014*; Chow, S.S.M., Liu, J.K., Hui, L.C.K., Yiu, S.M., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2014, Volume 8782. https://doi.org/10.1007/978-3-319-12475-9_1.
40. Taubert, T.; Wood, C.A. SPAKE2+, an Augmented Password-Authenticated Key Exch. (PAKE) Protocol. RFC 9383. September 2023. Available online: <https://www.rfc-editor.org/info/rfc9383/> (accessed on 8 September 2025).
41. Zeng, J.P.R. A Generic Construction of Tightly Secure Password-based Authenticated Key Exchange. In *International Conference on the Theory and Application of Cryptology and Information Security*; Springer: Singapore, 2023.
42. Malik, A.; Kushwah, R. A Secure Authentication and Key Establishment Protocol for IoT Devices Using ECCMD. *SN Comput. Sci.* **2025**, *6*, 803. <https://doi.org/10.1007/s42979-025-04354-x>.
43. Kumar, K.; Ahlawat, P. A Survey on Path Key Establishment. In *Cyber Security and Digital Forensics, REDCYSEC 2023*; Lecture Notes in Networks and Systems; Roy, N.R., Tanwar, S., Batra, U., Eds.; Springer: Singapore, 2024; Volume 896. https://doi.org/10.1007/978-981-99-9811-1_5.
44. Lopez Ramos, J.A.; Rosenthal, J.; Schipani, D.; Schnyder, R. Group key management based on semigroup actions. *J. Algebra Its Appl.* **2017**, *16*, 1750148.
45. Alexei, G.; Myasnikov; Shpilrain, V.; Ushakov, A. *Non-Commutative Cryptography and Complexity of Group-Theoretic Problems*; American Mathematical Society: Providence, RI, USA, 2011.
46. Kumar, G.; Saini, H. Novel Noncommutative Cryptography Scheme Using Extra Special Group. *Secur. Commun. Netw.* **2017**, *2017*, 9036382. <https://doi.org/10.1155/2017/9036382>.
47. Habeeb, M.; Kahrobaei, D.; Koupparis, C.; Shpilrain, V. Public Key Exchange Using Semidirect Product of (Semi)Groups. In *Applied Cryptography and Network Security, ACNS 2013*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2013; Volume 7954, pp. 475–486.
48. Moldovyan, D.N.; Moldovyan, N.A. A New Hard Problem over Noncommutative Finite Groups for Cryptographic Protocols. In *International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security, MMM-ACNS 2010*:

- Computer Network Security*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 183–194. https://doi.org/10.1007/978-3-642-14706-7_14.
49. Sakalauskas, L.; Tvarijonas, P.; Raulynaitis, A. Key Agreement Protocol (KAP) Using Conjugacy and Discrete Logarithm Problem in Group Representation Level. *INFORMATICA* **2007**, *18*, 115–124.
 50. Shpilrain, V.; Ushakov, A. The conjugacy search problem in public key cryptography: unnecessary and insufficient. *Appl. Algebra Eng. Commun. Comput.* **2006**, *17*, 285–289.
 51. Kahrobaei, D.; Khan, B. A non-commutative generalization of ElGamal key exchange using polycyclic groups. In *IEEE GLOBE-COM 2006—2006 Global Telecommunications Conference [4150920]*; IEEE: Piscataway, NJ, USA, 2006. <https://doi.org/10.1109/GLOCOM.2006>.
 52. Myasnikov, A.; Shpilrain, V.; Ushakov, A. *Group-Based Cryptography*; Birkhäuser Verlag: Berlin, Germany, 2008.
 53. Cao, Z. *New Directions of Modern Cryptography*; CRC Press, Taylor & Francis Group: Boca Raton, FL, USA, 2012.
 54. Fine, B. Aspects of Non abelian Group Based Cryptography: A Survey and Open Problems. *arXiv* **2011**, arXiv:1103.4093.
 55. Anshel, M.; Goldfeld, D. An algebraic method for public-key cryptography. *Math. Res. Lett.* **1999**, *6*, 169–173/287–291.
 56. Blackburn, S.R.; Galbraith, S.D. Cryptanalysis of two cryptosystems based on group actions. In *Advances in Cryptology-ASIACRYPT '99*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1999; Volume 1716, pp. 52–61.
 57. Ko, K.H.; Lee, S.J.; Cheon, J.H.; Han, J.W.; Kang, J.S.; Park, C. New publickey cryptosystem using braid groups. In *Advances in Cryptology-CRYPTO 2000, Santa Barbara, CA*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2000; Volume 1880, pp. 166–183.
 58. Maze, G.; Monico, C.; Rosenthal, J. Public key cryptography based on semigroup actions. *Adv. Math. Commun.* **2007**, *1*, 489–507.
 59. Kropholler, P.H.; Pride, S.J.; Othman, W.A.M.; Wong, K.B.; Wong, P.C. Properties of certain semigroups and their potential as platforms for cryptosystems. *Semigroup Forum* **2010**, *81*, 172–186.
 60. Roman'kov, V.A. A nonlinear decomposition attack. *Groups Complex Cryptol.* **2016**, *8*, 197–207.27.
 61. Roman'kov, V. An improved version of the AAG cryptographic protocol. *Groups Complex Cryptol.* **2019**, *11*, 35–42.
 62. Ben-Zvi, A.; Kalka, A.; Tsaban, B. Cryptanalysis via algebraic span. In *Advances in Cryptology-CRYPTO, Proceedings of the 2018 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, 19–23 August 2018*; Shacham, H., Boldyreva, A., Eds.; Proceedings, Part I; Springer: Cham, Switzerland, 2018; Volume 10991, pp. 255–274.
 63. Tsaban, B. Polynomial-time solutions of computational problems in noncommutative-algebraic cryptography. *J. Cryptol.* **2015**, *3*, 601–622.
 64. Roman'kov, V. Cryptanalysis of a new version of the MOR scheme. *arXiv* **2019**, arXiv:1911.00895.
 65. Roman'kov, V. Cryptanalysis of two schemes of Baba et al. by linear algebra methods. *arXiv* **2019**, arXiv:1910.09480.
 66. Ben-Zvi, A.; Kalka, A.G.; Tsaban, B. Cryptanalysis via Algebraic Spans. *CRYPTO* **2018**, *1*, 255–274.
 67. Noether, M. Luigi Cremona. *Math. Ann.* **1904**, *59*, 1–19.
 68. Shafarevich, I.R. On some infinite dimension groups II. *Izv. Akad. Sci. Ser. Math.* **1981**, *2*, 214–226.
 69. Bodnarchuk, Y. Every regular automorphism of the affine Cremona group is inner. *J. Pure Appl. Algebra* **2001**, *157*, 115–119.
 70. Machi, A. *Algebra for Symbolic Computations*; Springer: Berlin/Heidelberg, Germany, 2012; Volume VIII, 180p.
 71. Wroblewska, A. On some properties of graph based public keys. *Albanian J. Math.* **2008**, *2*, 229–234.
 72. Wróblewska, A. Lingwistyczne układy dynamiczne oparte na grafach algebraicznych i ich zastosowanie w kryptografii. Ph.D. Thesis, PAN Instytut Podstawowych Problemów Techniki, Warszawa, Poland, 2016. Available online: https://www.ippt.pan.pl/repository/doktoraty/open/2016wroblewska_a_doktorat.pdf (accessed on 8 September 2025).
 73. Ustimenko, V.; Klisowski, M. On new protocols of noncommutative cryptography in terms of homomorphism of stable multivariate transformation groups. *J. Algebra Discret. Math.* **2023**, 220–250. <https://doi.org/10.12958/adm1523>.
 74. Ustimenko, V. On Desynchronised Multivariate Algorithms of El Gamal Type for Stable Semigroups of Affine Cremona Group. *Theor. Appl. Cybersecur.* **2019**, *1*, 22–30.
 75. Ustimenko, V. On new symbolic key exchange protocols and cryptosystems based on hidden tame homomorphism. *Proc. Rep. Math. Acad. Sci. Ukr.* **2018**, *10*, 26–36.
 76. Bollobash, B. *Extremal Graph Theory*; Academic Press: London, UK, 1978.
 77. Biggs, N. *Algebraic Graphs Theory*, 2nd ed.; Cambridge University Press: Cambridge, UK, 1993.
 78. Lubotzky, A.; Phillips, R.; Sarnak, P. Ramanujan graphs. *Combinatorica* **1988**, *8*, 261–277. <https://doi.org/10.1007/BF02126799>.
 79. Paar, C.; Pelzl, J. Key Establishment. In *Understanding Cryptography*; Springer: Berlin/Heidelberg, Germany, 2010. https://doi.org/10.1007/978-3-642-04101-3_13
 80. FIPS 203; Module-Lattice-Based Key-Encapsulation Mechanism Standard. NIST Federal Information Processing Standards: Gaithersburg, MD, USA, 2024. Available online: <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.203.pdf> (accessed on 8 September 2025).
 81. Bradner, S. Key Words for Use in RFCs to Indicate Requirement Levels. BCP 14, RFC 2119. March 1997. Available online: <https://www.rfc-editor.org/rfc/rfc2119> (accessed on 8 September 2025).

82. Kaufman, C.; Hoffman, P.; Nir, Y.; Eronen, P.; Kivinen, T. Internet Key Exchange Protocol Version 2 (IKEv2). STD 79, RFC 7296. October 2014. Available online: <https://www.rfc-editor.org/rfc/rfc7296> (accessed on 8 September 2025).
83. Leiba, B. Ambiguity of Uppercase vs Lowercase in RFC; BCP 14, RFC 8174. May 2017. Available online: <https://www.rfc-editor.org/rfc/rfc8174> (accessed on 8 September 2025).
84. Smyslov, V. Intermediate Exchange in the Internet Key Exchange Protocol Version 2 (IKEv2). RFC 9242. May 2022. Available online: <https://www.rfc-editor.org/rfc/rfc9242> (accessed on 8 September 2025).
85. Tjhai, C.J.; Tomlinson, M.; Bartlett, G.; Fluhrer, S.; Van Geest, D.; Garcia-Morchon, O.; Smyslov, V. Multiple Key Exchanges in the Internet Key Exchange Protocol Version 2 (IKEv2). RFC 9370. May 2023. Available online: <https://www.rfc-editor.org/rfc/rfc9370> (accessed on 8 September 2025).
86. NIST SP 800-56Ar3; Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography. NIST Special Publication (SP). Department of Commerce: Washington, DC, USA, 2018. <https://doi.org/10.6028/NIST.SP.800-56Ar3>.
87. NIST SP 800-56Br2; Recommendation for Pair-Wise Key-Establishment Using Integer Factorization Cryptography. NIST Special Publication (SP). Department of Commerce: Washington, DC, USA, 2019. <https://doi.org/10.6028/NIST.SP.800-56Br2>.
88. NIST FIPS 203; Module-Lattice-Based Key Encapsulation Mechanism Standard. Federal Information Processing Standards Publications (FIPS). Department of Commerce: Washington, DC, USA, 2024. <https://doi.org/10.6028/NIST.FIPS.2039>.
89. NIST IR 8547ipd; Transition to Post Quantum Cryptography Standards. NIST Internal Report (IR). Department of Commerce: Washington, DC, USA, 2024. <https://doi.org/10.6028/NIST.IR.8547.ipd>.
90. NIST SP 800-90C; Recommendation for Random Bit Generator (RBG). NIST Special Publication (SP). National Institute of Standards and Technology: Gaithersburg, MD, USA, 2025. <https://doi.org/10.6028/NIST.SP.800-90C>.
91. NIST SP 800-152; A Profile for U.S. Federal Cryptographic Key Management Systems. NIST Special Publication (SP). National Institute of Standards and Technology: Gaithersburg, MD, USA, 2015. <https://doi.org/10.6028/NIST.SP.800-15244>.
92. Brainard, J.; Kaliski, B.; Turner, S.; Randall, J. Use of the RSA-KEM Key Transport Algorithm in the Cryptographic Message Syntax (CMS). RFC 5990. September 2010. Available online: <https://www.rfc-editor.org/info/rfc5990/> (accessed on 8 September 2025).
93. ANSI X9.63-2011; (R2017)—Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography. Accredited Standards Committee X9: Annapolis, MD, USA, 2011. Available online: <https://webstore.ansi.org/standards/ascx9/ansix9442007r2017> (accessed on 8 September 2025).
94. NIST SP 800-133r2; Recommendation for Cryptographic Key Generation. NIST Special Publication (SP). Department of Commerce: Washington, DC, USA, 2020. <https://doi.org/10.6028/NIST.SP.800-133r2>.
95. NIST SP 800-57pt1r5; Recommendation for Key Management: Part 1—General. NIST Special Publication (SP). National Institute of Standards and Technology: Gaithersburg, MD, USA, 2020. <https://doi.org/10.6028/NIST.SP.800-57pt1r5>.
96. NIST SP 800-108r1-upd1; Recommendation for Key Derivation Using Pseudorandom Functions. NIST Special Publication (SP). Includes Updates as of 2 February 2024. National Institute of Standards and Technology: Gaithersburg, MD, USA, 2022. <https://doi.org/10.6028/NIST.SP.800-108r1-upd1>.
97. NIST SP 800-56Cr2; Recommendation for Key-Derivation Methods in Key-Establishment Schemes. NIST Special Publication (SP). Department of Commerce: Washington, DC, USA, 2020. <https://doi.org/10.6028/NIST.SP.800-56Cr2>.
98. Barbosa, M.; Connolly, D.; Duarte, J.D.; Kaiser, A.; Schwabe, P.; Varner, K.; Westerbaan, B. X-Wing: The Hybrid KEM You’ve Been Looking For. *IACR Commun. Cryptol.* **2024**, *1*, 1–22. <https://doi.org/10.62056/a3qj89n4e>.
99. Avanzi, R.; Bos, J.; Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schanck, J.M.; Schwabe, P.; Seiler, G.; Stehlé, D. CRYSTALS-Kyber Algorithm Specifications and Supporting Documentation. *NIST PQC Round* **2019**, *2*, 1–43.
100. Almeida, J.B.S.; Olmos, A.; Barbosa, M.; Barthe, G.; Dupressoir, F.; Grégoire, B.; Laporte, V.; Lécenet, J.C.; Low, C.; Oliveira, T.; et al. Formally verifying Kyber Episode V: Machine-checked IND-CCA Security and Correct Ness of ML-KEM. *Cryptology ePrint Archive*, Paper 2024/843. Available online: <https://eprint.iacr.org/2024/843> (accessed on 8 September 2025).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.