On the Generator of Stable Cubical Multivariate Encryption Maps Over Boolean Rings for Protection of Large Information System

Vasyl Ustimenko
Institute of Mathematics, Maria Curie-Skldowska University, Lublin

Michał Klisowski
Institute of Computer Science, Maria Curie-Skldowska University, Lublin

December 27, 2021

Abstract

Encryption based on Walks in Algebraic GRAphs (EWAGRA) is used for protectionof authors' rights, access to electronic books or documents located at a certain knowledge base (Information Quality Assurance Support Systems of a university, digital library supporting distance education, various digital archives and etc). The method allows generating nonlinear stream ciphers, which have some similarity with a one-time pad: different keys produce distinct ciphertexts from the same plaintext. In contrast to the case of a one-time pad, the length of the key is flexible and the encryption map is a nonlinear polynomial map, which order is growing with the growth of the dimension n of the plaintext space. The encryption has good resistance to attacks of the adversary when he has no access to plaintext space or has a rather small number of intercepted plaintext-ciphertext pairs. It is known that encryption and decryption maps are cubical maps. So, interception of $n^3 + O(n)$ plaintext-ciphertext pairs allows conducting a plain linearization attack for finding the inverse map. We consider the idea of modification of this encryption algorithm after sending each message without use key exchange protocols. So the new algorithm is resistant to plain linearization attacks.

Keywords: stream ciphers, key exchange protocols, audentification, graph based, encryption, multivariate maps

1 Introduction

Extremal graph-based methods of encryption are already used to protect Information Systems. The general idea is to treat vertices of algebraic graphs as messages and walks of a certain length as encryption tools. The quality of such private key encryption is good for graphs defined over finite fields, which sizes are close to the bound of Even Cycle Theorem formulated by Erdős (see [1, 2, 3]). As it was recently found new construction of algebraic graphs obtained simply by the change of finite field $\mathbb{F}_q$ for a finite commutative ring K can also be used for effective encryption. For practical implementation cases of "fast" rings (modular arithmetic, Boolean rings, etc.) can be useful for the development of fast graph-based encryption algorithms working on plaintext space K_n with the use of key space M_s, where M is a multiplicatively closed subset of nonzero elements from K and s is restricted from above by linear function $l(n)$. As in the case of one-time pad different passwords produce distinct ciphertexts. In contrast to the case of a one-time pad the length of a password $2 \leq s \leq l(n)$ is flexible and the multivariate encryp-

tion map is nonlinear.

It allows in practice to use a stream cipher, which has good resistance to attacks of the adversary without access to plaintext space.

Let us discuss the resistance to plaintext-ciphertext attacks for the graph-based algorithm, when an encryption map corresponds to the chosen path.

A member of the family of k-regular algebraic graphs G_n, $n \geq 2$, where k, $k > 2$ is constant, of increasing order and girth gives us a finite approximation of infinite k-regular tree.

The idea of security for the walk based algorithm based on the fact that finding a pass between two given vertices at the distance d of infinite k-regular tree (or k-regular graph of girth $> d$) require $f(k,d) = k(k-1)^{d-1}$ steps of the natural branching process. If the girth of the graph is unbounded and distance d can be unknown the problem is getting harder. The complexity $f(k,d)$ is growing when d is increasing. Obviously, the pass between two vertices can be found by the Dijkstra algorithm with the complexity $v\ln(v)$, where v is the order of the graph. In the case of our family the order of a graph representative is growing exponentially, so the Dijkstra algorithm is not applicable.

For some classes of the graphs these encryption algorithms have also some resistance to plaintext-ciphertext attacks when the number of known plaintext-ciphertext pairs is restricted.

The last feature allows using walks on special algebraic graphs for the establishment of key exchange protocols, authentication algorithms (electronic signatures) and development of hash functions.

It turns out that some classes of such graph-based encryption maps form a family of stable multivariate encryption maps of large order, i.e. maps generating a large group of transformations of bounded polynomial degree. In such a case the degrees of multivariate encryption and decryption bijective maps are restricted by a constant. It means that their resistance to plaintext-ciphertext attack is bounded by the option of linearization cryptanalysis. We need a polynomial number of plaintext-ciphertext pairs to conduct a successful cryptanalytic attack in polynomial time $P(n)$. Notice that if the degree of $P(n)$ is "sufficiently large" such attacks are not feasible.

In this paper, we are not going to develop EWA-GRA theory for rather wide classes of algebraic graphs (like linguistic graphs or graphs related to time dependent dynamical systems). First steps in graph based encryption were connected with applications of the family of algebraic graphs of large girth $D(n,q)$, which had been used for the development of known class of important LDPC codes, and their generalizations $D(n,K)$ (see [4, 5, 6]). We concentrate on further research on generation of encryption maps related to walks on these graphs and graphs $A(n,K)$ (special homomorphic images of $D(n,K)$) in a special case of Boolean ring $K = \mathbb{B}(m)$. We hope that implementations of such algorithms can be used for the protection of public Information and Management Systems such as e-parliament, University Quality Assurance Support Systems, GIS, various e-governing special systems. The mentioned above term "level of security" is defined only by modulo of possible cryptanalysis. We simply assume that the growth of degree and order of multivariate map may lead to a better level of security.

2 Towards Applied K-theory of Algebraic Graphs, Case of Boolean Rings

Notice, that mentioned above families of graphs $D(n,K)$ and $A(n,K)$ are defined over general commutative ring K. Studies of the case of finite fields were conducted on the theoretical level and via computer simulation. They brought many constructive results of Extremal Graph Theory some practically used algorithm. The case of arithmetical rings, especially rings $\mathbb{Z}_{2^7}$, $\mathbb{Z}_{2^8}$, $\mathbb{Z}_{2^{16}}$, $\mathbb{Z}_{2^{32}}$ and $\mathbb{Z}_{2^{64}}$, was partially investigated via computer implementations. Of course, the change of finite fields for computer arithmetic makes corresponding algorithms faster. Obviously, there are many other interesting "fast" rings. Among them, we have Boolean rings $K = \mathbb{B}(m)$, which consist of all subsets of $\{1,2,\ldots,m\}$ with addition operation $A + B = A \cup B - A \cap B$ (symmetric difference of subsets A and B) and multiplication $A \times B = A \cap B$. We select the multiplicative subset

$M = M_i = \{A | i \in A\}$ as the alphabet for strings from key space. This subset is an analog of a subset $\mathbb{F}_q^*$ of $\mathbb{F}_q$.

In our paper, we present the first results of the implementation of cryptographical algorithms based on algebraic graphs defined over Boolean rings. We compare the evaluation of algorithms in the cases of $\mathbb{B}(m)$ and $\mathbb{F}_{2^m}$.

In the next section the reader can find graph-theoretical definitions, which we use in the description of algorithms, and some known results of Extremal Graph theory useful for the evaluation of encryption properties.

Section 4 contains the description of the families of algebraic graphs $D(n, K)$ and $A(n, K)$ defined over general commutative ring K.

In Section 5 special transformation groups related to graphs $A(n, K)$ and $D(n, K)$ are introduced. We formulate results on their stability (all transformations from the group are cubical). The symmetric stream cipher which uses mentioned above transformations is described and some theoretical properties of such encryption are formulated. We introduce the key exchange protocol based on the complexity of the discrete logarithm problem for a cyclic subgroup of our group of stable transformation. Additionally, we consider the method of symmetric multivariate encryption in a multiuser mode, which can be used for different problems of access control.

The weak side of symmetric algorithms (in numerical and multivariate mode) is an option to conduct cubical linearization cryptographical attacks because the inverse map is also cubical. We demonstrate that the use of classical Diffie-Hellman protocol based on cyclic subgroup $\mathbb{Z}_p^*$, where p is prime, can protect users against such attacks. Notice that described in this section algorithms may be used in the case of other stable subgroups of the Cremona group. We further discuss the first implementation of such algorithms in the case of Boolean rings $\mathbb{B}(m)$ consisting of 2^m ring elements.

In the appendix, we present the results of computer simulation: time evaluation of the basic private key algorithm, time evaluation for the generation of core cubical maps, which can be used for multivariate symmetric encryption and key exchange protocols. We compute the densities of generated cubical maps. Conclusions are given in section 7.

3 Some Definitions from Graph Theory

The missing definitions of graph-theoretical concepts in case of simple graphs which appears in this paper can be found in [7, 1].

All graphs we consider are *simple graphs*, i.e. undirected, without loops and multiple edges. Let $V(\Gamma)$ and $E(\Gamma)$ denote the set of vertices and the set of edges of Γ respectively. $|V(\Gamma)$ and $E(\Gamma)|$ is called the *order* of Γ, and $|E(\Gamma)|$ is called the *size* of G. A path in Γ is called *simple path* if all its vertices are distinct. We shall identify Γ with the corresponding anti-reflexive binary relation on $V(\Gamma)$, i.e. $E(\Gamma)$ is a subset of $V(\Gamma) \times V(\Gamma)$. A graph Γ is *bipartite* if none of its two vertices belonging to the same set are in the relationship (i.e. if its vertices can be partitioned into two sets in such a way that no edge joins two vertices in the same set). The *length of a path* is a number of its edges. The *girth* of a graph Γ, denoted by $g = g(\Gamma)$ is the length of the shortest cycle in Γ.

We refer to the family of regular simple graphs Γ_i of degree k_i and order v_i as a *family of graphs of increasing girth* if

$$g(\Gamma_i) \geq g(\Gamma_{i-1})$$

Recall, that family of regular graphs Γ_i of degree k_i and increasing order v_i is a *family of graphs of large girth* if

$$g(\Gamma_i) \geq c \log_{k_i}(v_i),$$

for some independent constant c, $c > 0$ (see [8, 7, 1]). These families play an important role in Extremal Graph Theory, Theory of LDPC codes (see [9]) and Cryptography (see [2, 3] and further references). The family of graphs of large girth of bounded degree are hard to construct. This fact is a serious motivation for studies of infinite families of graphs of increasing girth, which are generalizations of families of graphs of large girth.

Just three explicit constructions of families of graphs of large girth for $k_i = k$, $i = 1, 2, \ldots$ (k is

independent but unbounded constant) for the general case of arbitrary large k with unbounded girth are known: the family of Cayley-Ramanujan graphs introduced by G. Margulis (see [10] and [11]), where girth was computed and spectral gap was evaluated), which appeared approximately forty years after the publication of Erdős probabilistic construction (see [12, 1] and further references), the family of algebraic graphs $D(n,q)$ with connected components $CD(n,q)$ defined over the arbitrary finite field $\mathbb{F}_q$ (see [6] or [4], [5], where these graphs are used for the construction of LDPC codes), and regular versions of polarity graphs for $D(n,q)$ or $CD(n,q)$.

Below we define the family of graphs $A(n,q)$ of increasing girth. This family of algebraic graphs is not edge-transitive, so the problem of evaluation of girth for its members is difficult.

4 The Algebraic Graphs $A(N,K)$ and $D(N,K)$

Below we consider the family of graphs $A(n,K)$ ($D(n,K)$, respectively) where $n > 5$ is a positive integer and K is a commutative ring. In case of $K = \mathbb{F}_q$ we denote $A(n,q)$ ($D(n,q)$, respectively).

Let P and L be two copies of Cartesian power $K^{\mathbb{N}}$, where K is the commutative ring and $\mathbb{N}$ is the set of positive integer numbers. Elements of P will be called *points* and those of L *lines*. To distinguish points from lines we use parentheses and brackets. If $x \in V$, then $(x) \in P$ and $[x] \in L$. It will also be advantageous to adopt the notation for coordinates of points and lines for the case of a general commutative ring K:

$$(p) = (p_{0,1}, \ldots, p_{i,i}, p_{i,i+1}, \ldots),$$
$$[l] = [l_{1,0}, \ldots, l_{i,i}, l_{i,i+1}, \ldots]$$

$$\Big((p) = (p_{0,1}, p_{1,1}, p_{1,2}, p_{2,1}, \ldots, p_{i,i}, p'_{i,i}, p_{i,i+1}, p_{i+1,i}, \ldots),$$
$$[l] = [l_{1,0}, l_{1,1}, l_{1,2}, l_{2,1}, \ldots, l_{i,i}, l'_{i,i}, l_{i,i+1}, l_{i+1,i}, \ldots]\Big).$$

The elements of P and L can be thought as infinite ordered tuples of elements from K, such that only a finite number of components are different from zero. We now define an incidence structure (P, L, I) as follows. We say the point (p) is incident with the line $[l]$, and we write $(p)I[l]$, if the following relations between their coordinates hold:

$$\begin{array}{ll} l_{1,1} - p_{1,1} = l_{1,0}p_{0,1} & l_{i,i} - p_{i,i} = l_{1,0}p_{i-1,i} \\ l_{1,2} - p_{1,2} = l_{1,1}p_{0,1} & l'_{i,i} - p'_{i,i} = l_{i,i-1}p_{0,1} \\ l_{2,1} - p_{2,1} = l_{1,0}p_{1,1} & l_{i,i+1} - p_{i,i+1} = l_{i,i}p_{0,1} \\ & l_{i+1,i} - p_{i+1,i} = l_{1,0}p'_{i,i} \end{array} \quad (1)$$

For each positive integer $n \geq 2$ we obtain an incidence structure (P_n, L_n, I_n) as follows. First, P_n and L_n are obtained from P and L, respectively, by simply projecting each vector into its n initial coordinates with respect to the above order. The incidence I_n is then defined by imposing the first n–1 incidence equations and ignoring all others. The incidence graph corresponding to the structure (P_n, L_n, I_n) is denoted by $A(n,K)$ ($D(n,K)$, respectively).

For each positive integer $n \geq 2$ we consider the *standard* graph homomorphism ϕ_n of (P_n, L_n, I_n) onto $(P_{n-1}, L_{n-1}, I_{n-1})$ defined as simple projection of each vector from P_n and L_n onto its $n-1$ initial coordinates with respect to the above order.

In the case of $K = \mathbb{F}_q$ we write $D(n,q)$ and $A(n,q)$ instead of $D(n,K)$ and $A(n,K)$

We define the *colour function* π for the graph as a projection of tuples $(p) \in P_n$ and $[l] \in L_n$ onto the first coordinate (p) or $[l]$, respectively. So the set of colours is $\mathbb{F}_q$. We assume that $N_\alpha(v)$ is the *operator of taking the neighbour of* v *of colour* $\pi(v) + \alpha$ in our graph.

5 On Transformation Groups Related to Algebraic Graphs and Their Direct Cryptographical Applications

Cryptograpical basics and results on complexity the reader can find in [13] and [6].

Let $G_n = G_n(K)$ be the group of transformations of variety $P \cup L$ ($\mathbb{F}_q^n \cup \mathbb{F}_q^n$) with generators N_α ($\alpha \in K$). Let M be a multiplicative subset in the ring K i.e. closed under multiplication subset without 0 element.

Theorem 1.

- $\lim_{n\to\infty} |G_n| = \infty$,
- *$g \in G_n$ is a cubical map.*
- *Let M be a multiplicative subset of K and for each i element $\alpha_i + \alpha_{i+1}$ belongs to M and $\alpha_1 + \alpha_n \in M$. Then the order of $g_n = N_{\alpha_1} N_{\alpha_2} \dots N_{\alpha_s}$, $g_n \neq e$, is going to ∞ with the growth of parameter n.*

We say g is *cubical map* if it has a form $g = (f_1(x_1, \dots, x_n), \dots, f_n(x_1, \dots, x_n))$, where $y_i = f_i(x_1, \dots, x_n)$ are polynomials of n variables written as the sums of monomials of kind $x_{i_1}^{n_1} x_{i_2}^{n_2} x_{i_3}^{n_3}$, for $i_1, i_2, i_3 \in 1, 2, \dots, n$; $n_1,\ n_2,\ n_3 \in \{0, 1, 2, 3\}$, $n_1 + n_2 + n_3 \leq 3$ with the coefficients from $K = \mathbb{F}_q$. As we mention before the polynomial equations $y_i = f_i(x_1, x_2, \dots, x_n)$ have the degree 3.

The security of some of the algorithms presented below is based on the complexity of the discrete logarithm problem for the group $G_n = G_n(K)$.

5.1 On Some Cryptographical Algorithms

The plaintext space of the algorithm is the affine variety K^n, where K is the chosen commutative ring. Graph theoretical encryption corresponds to walk on the bipartite graph with partition sets which are isomorphic to K^n. We conjugate chosen graph-based encryption map, which is a composition of several elementary cubical polynomial automorphisms of a free module K^n with special invertible affine transformation of K^n. Finally, we compute symbolically the corresponding cubic public map g of K^n onto K^n. We evaluate time for the generation of g, and the number of monomial expression in the list of corresponding public rules. Let $M(K) \subset K \setminus \{0\}$, which is closed under multiplication.

A_1: Private-key symmetric algorithm

We assume that two users Alice and Bob, share a common password consisting the sequence of colors $\alpha_1, \alpha_2, \dots, \alpha_s$, where $\alpha_i + \alpha_{i+1} \in M(K)$, $i = 1, \dots, s-1$ and two affine transformations τ_1, τ_2 form affine group $AGL(n, K)$. Then, they encrypt the plaintext p to ciphertext c as follows: $c = \tau_1 N_{\alpha_1} N_{\alpha_2} \dots N_{\alpha_s} \tau_2(p)$. Decryption process is as follows: $p = \tau_2^{-1} N_{-\alpha_s} N_{-\alpha_{s-1}} \dots N_{-\alpha_1} \tau_1^{-1}(c)$.

If $s < \frac{g(A(n,q))}{2}$ ($s < \frac{g(D(n,q))}{2}$), then different keys produce distinct ciphertext.

It is clear that users may choose key exchange protocol and change or modify maps τ_1, τ_2 and string $\alpha_1, \alpha_2, \dots, \alpha_s$. This measure gives protection against possible linearization attacks. In the next section, we consider the option of serious modification of the encryption map without key exchange protocols.

A_2: Private-key algorithm for multiusers network

We assume that $\alpha_i + \alpha_{i+1} \in M(K)$ for $i = 1, 2, \dots$. Alice takes τ_1, $\tau_2 = {\tau_1}^{-1}$, sequence $\alpha_1, \alpha_2, \dots, \alpha_s$ and forms the map $f_B = \tau_1 N_{\alpha_1} N_{\alpha_2} \dots N_{\alpha_s} \tau_2$ in a symbolic way (She can use Maple, SageMath or Mathematica). Here B stands for the pair b $= (\alpha_1, \alpha_2, \dots, \alpha_s)$, τ_1. She gets the encryption map as a cubical public rule: $x_1 \to f_1(x_1, x_2, \dots, x_n)$, $x_2 \to f_2(x_1, x_2, \dots, x_n)$, $\dots, x_n \to f_n(x_1, x_2, \dots, x_n)$, where f_i are multivariable polynomials from $K[x_1, x_2, \dots, x_n]$. For simplicity, we assume that s is even.

Let $S_k = (B_k, J_k)$, $k = 1, 2, \dots, N$ be the pairs of users (B and J stand for Brad and Jennifer). Alice provides each pair with the "seed" triple C_k, f_{B_k}, D_k, where C_k and D_k are linear or affine transformations of the plainspace K^n of large order (like maps conjugated with Singer cycles of order $q^n - 1$ in the case of $K = \mathbb{F}_q$, see [14] or [15]) and also gives them ${f_{B_k}}^{-1}$. So they can use encryption map $C_k f_{B_k} D_k$ and decrypt with ${D_k}^{-1} f_{B_k}^{-1} {C_k}^{-1}$

The pair S_k can take "closest possible" primes p_1, p_2, p_3 (or pseudoprimes) numbers to $p_1 = |C_k|$, $p_2 = |D_k|$ and $|f_{B_k}|$. They use Diffie-Hellman key exchange protocol for ${\mathbb{Z}_{p_i}}^*$ and develop collision triple

$h_i \in {\mathbb{Z}_{p_i}}^*$, $i = 1, 2, 3$. During the session they use encryption and decryption cubical maps ${C_k}^{h_1} {f_{B_k}}^{h_2} D_k^{h_3}$ and ${D_k}^{-h_3} {f_{B_k}}^{-h_2} D_k^{-h_1}$.

Notice that S_k is known to trusted third party (Alice), but triple h_1, h_2, h_3 is individual private password for Brad and Jennifer. There is no need to compute a new encryption map symbolically, users just apply $D_k^{h_3}$, $f_{B_k}^{h_2}$ and ${C_k}^{h_1}$ consecutively to plaintext space vector. If the next session of the key exchange Brad and Jennifer can get new triple $h'_j \in {\mathbb{Z}_{p_j}}^*$, $j = 1, 2, 3$ and use numbers $h"_j = h'_j h_j \bmod p_j$ for the modification of the multivariate encryption map. This approach leads to dependence of the algorithm from the prehistory of communications.

The use of key exchange protocols as tools of protection against linearization attacks is the standard one. In the next section, we consider the similar algorithm of modification of encryption rule without key exchange.

A_3: Symbolic Diffie-Hellman algorithm

Suppose Alice and Bob want to agree on a key K_{AB}.

1. The first step Alice computes $f = \tau_1 N_{\alpha_1} N_{\alpha_2} \dots N_{\alpha_s} \tau_1^{-1}$ ($\alpha_i + \alpha_{i+1} \in M(K)$, $i = 1, \dots, s-1$, $\alpha_1 + \alpha_s \in M(K)$) of large order with usage of graph $A(n, K)$ or $D(n, K)$ and she sends f to Bob. The next step is for Alice to pick a secret integer n_A that she does not reveal to anyone, while at the same time Bob picks an integer n_B that he keeps secret.

2. Bob and Alice use their secret integers (n_A and n_B, respectively) to compute $A = f^{n_A}$ and $B = f^{n_B}$, respectively. They use composition of multivariable map f with itself. They exchange these computed values.

3. Finally, Alice and Bob again use their secret integers to compute $K_{AB} \equiv B^{n_A} \equiv (f^{n_B})^{n_A} = f^{n_A n_B}$, and $K_{AB} \equiv A^{n_B} \equiv (f^{n_A})^{n_B} = f^{n_A n_B}$, respectively.

In the previous algorithm, Alice and Bob may use the symbolic Diffie Hellman protocol instead of the classical one. They also can use key exchange protocols for cyclic groups generated by matrices C_k and D_k of large order.

6 Connected Components of $D(N, K)$ and Modernized Encryption Without Key Exchange

For studies of connected components of graphs $D(n, K)$, where K is a finite commutative ring, it will be convenient for us to define

$$
\begin{aligned}
&p_{-1,0} = l_{0,-1} = p_{1,0} = l_{0,1} = 0,\\
&p_{0,0} = l_{0,0} = -1,\\
&p'_{0,0} = l'_{0,0} = 1,\\
&p_{0,1} = p_2,\\
&l_{1,0} = l_1,\\
&l'_{1,1} = l_{1,1},\\
&p'_{1,1} = p_{1,1},
\end{aligned}
$$

and to rewrite (1) in the form:

$$
\begin{aligned}
&l_{ii} - p_{ii} = l_1 p_{i-1,i}\\
&l'_{ii} - p'_{ii} = l_{i,i-1} p_1\\
&l_{i,i+1} - p_{i,i+1} = l_{ii} p_1\\
&l_{i+1,i} - p_{i+1,i} = l_1 p'_{ii}
\end{aligned}
\qquad (2)
$$

for $i = 0, 1, 2, \dots$

Notice that for $i = 0$, the four conditions (2) are satisfied by every point and line, and, for $i = 1$, the first two equations coincide and give $l_{1,1} - p_{1,1} = l_1 p_1$.

For each positive integer $k \geq 2$ we obtain an incidence structure (P_k, L_k, I_k) as follows. First, P_k and L_k are obtained from P and L, respectively, by simply projecting each vector onto its k initial coordinates. Then the incidence I_k is defined by imposing the first $k-1$ incidence relations and ignoring all others. For fixed q, the incidence graph corresponding to the structure (P_k, L_k, I_k) is denoted by $D(k, K)$. In case $K = \mathbb{F}_q$ we assume that $D(k, q) = D(k, \mathbb{F}_q)$. It is convenient to define $D(1, q)$ to be equal to $D(2, q)$. The properties of the graphs $D(k, q)$ that we are concerned with are described in the following Proposition.

Proposition 1. *Let q be a prime power, and $k \geq 2$. Then*

(i) $D(k,q)$ is a q-regular bipartite graph of order $2q^k$;

(ii) for odd k, $g(D(k,q)) \geq k+5$;

(iii) for odd k and $q \equiv 1 \pmod{\frac{k+5}{2}}$, $g(D(k,q)) = k+5$.

Let $k \geq 6$, $t = [(k+2)/4]$, and let

$$u = (u_i, u_{11}, \cdots, u_{tt}, u'_{tt}, u_{t,t+1}, u_{t+1,t}, \cdots)$$

be a vertex of $D(k,K)$. (It does not matter whether u is a point or a line). For every r, $2 \leq r \leq t$, let

$$a_r = a_r(u) = \sum_{i=0,m} (u_{ii}u'_{r-i,r-i} - u_{i,i+1}u_{r-i,r-i-1}),$$

and $a = a(u) = (a_2, a_3, \cdots, a_t)$. (Here we define $p_{0,-1} = l_{0,-1} = p_{1,0} = l_{0,1} = 0$, $p_{00} = l_{00} = -1$, $p_{0,1} = p_1$, $l_{1,0} = l_1$, $l'_{11} = l_{11}$, $p'_{1,1} = p_{1,1}$).

Proposition 2. . *Let u and v be vertices from the same component of $D(k,K)$. Then $a(u) = a(v)$. Moreover, for any $t-1$ ring elements $x_i \in K$, $2 \leq it \geq [(k+2)/4]$, there exists a vertex v of $D(k,K)$ for which*

$$a(v) = (x_2, \ldots, x_t) = (x).$$

Let us consider the following equivalence relation τ : $u\tau v$ iff $a(u) = a(v)$ on the set $P \cup L$ of vertices of $D(k,q)$ ($D(q)$).

Let $CD(n,q)$ be the connected component of $D(k,q)$ which contains $(0,0,\ldots)$. Let τ' be an equivalence relation on $V(D(k,K))$ ($D(K)$) such that the equivalences classes are the totality of connected components of this graph. According to previous propositions, $u\tau v$ implies $u\tau' v$. If char $GF(q)$ is an odd number, the converse of the last proposition is true.

Proposition 3. *Let q be an odd number. Vertices u and v of $D(q)$ ($D(k,q)$) belong to the same connected component iff $a(u) = a(v)$, i.e., $\tau = \tau'$.*

A_4: Modified private-key algorithm for multi-user network

Let us consider an option of changing the encryption map of algorithm A_2 in the case of graph $D(n,K)$. Recall that Alice provides Brad and Jennifer with the "seed" triple C, f_B, D, where C and D are linear or affine transformations of the plaintext space K^n of large order and gives them also ${f_{B_k}}^{-1}$. So they can use encryption map Cf_BD and decrypt with $D^{-1}f_B^{-1}C^{-1}$

Like in the case of modification with key exchange protocol Brad and Jennifer have to take the "closest possible" primes p_1, p_2, p_3 (or pseudoprimes) numbers to $p_1 = |C|, p_2 = |D|$ and $|f_B|$. Recall that $f_B = B^{-1}N_{\alpha_1}N_{\alpha_2}\ldots N_{\alpha_s}B$, where B is affine transformation of K^n.

We assume that string α_i, $i = 1,2,\ldots,s$ satisfies condition of Theorem 1. So the order of f_B is large.

Additionally, Alice chose functions $h_i(z_1, z_2, \ldots, z_{t-1})$, $t = [n + 1/4]$, which map K^t into $\mathbb{Z}^*_{p_i}$ and creates $h_i(x_1, x_2, \ldots, x_n) = h_i(a_2(B\mathrm{x}), a_3(B\mathrm{x}), \ldots, a_t(B\mathrm{x})) = \beta_i$.

Let a be any positive number and $Cf_B{}^a(D(x)) = y$, $C(B^{-1}(N^a(B(D(x))))) = y$. So $B(c^{-1}(y) = N^a(B(D(x))$. The function N^a corresponds to transition via pass in the graph $D(n,K)$. It means that vectors $B(c^{-1}(y))$ and $B(D(x))$ are from the same connected component. $a_i(B(D(x))) = a_i(B(C^{-1})(y))$ for each $i : 2 \leq i \leq t$.

It is important for us that $h_j(D(x)) = h_j(C^{-1}(y)) = \beta_j$, $j = 1,2,3$

So Brad computes $\beta_j(x)$ as left-hand side of the above equation.

He encrypts a plaintext x with $Cf_B{}^{\beta_2}D$ and sends it to Jennifer. She can compute β_2 as right-hand side of the above equation and decrypt with the $D^{-1}f_BC^{-1}$.

After the end of session Brad and Jennifer may change C for $C' = C^{\beta_1}$, D for $D' = D^{\beta_3}$ and functions $h_j(x_1, x_2, \ldots, x_n)$ for $h'_j(x_1, x_2, \ldots, x_n)^{\beta_j}$. During next session with the plaintext $(x'_1, x'_2, \ldots, x'_n)$ encryption will be with $C'f_B{}^{\beta_2\beta'_2}D'$, where $\beta'_2 = h'_2(D'(\mathrm{x}'))$.

7 Conclusion

In this paper, we present cryptographical algorithms based on families of graphs defined over boolean rings $K = B(m)$. In all cases, the basic maps are cubical multivariate bijective maps of the n-dimensional affine space over K. The set of algorithms consists of the family of fast stream ciphers implemented on the numerical level, protocols of key exchange and families of symbolic private key algorithms based on "seed" cubical multivariate maps given in the symbolic form (list of monomial summands given in a special order). Some properties of these algorithms are supported by mathematical statements, while some others simply supported by computer simulations. We present the important parameters of computer simulations such as execution times of numerical encryption algorithms, time required for the generation of "seed" multivariate cubical map and their comparison with similar graph-based algorithms defined over finite fields. The security of modernized stream ciphers (with eh option to iterate the "seed" map implemented on the "numerical" or "symbolic" level) and proposed key exchange protocols are based on the complexity of the discrete logarithm problem (DLP) for a cyclic subgroup of the Cremona group. Notice that our method produces a large group of cubic maps, so all elements of cyclic subgroups are cubical transformations. It means that the adversary is not able to use the order function to reduce a search for the DLP solution. The order of the chosen cyclic group is growing with the growth of the plaintext space dimension. If the dimension is "sufficiently large" then the computation of order is not feasible for an adversary. In the case of the Boolean ring and other commutative rings of characteristic 2, the order of the cyclic group generated by our nonlinear map has to be a power of two. We have to mention that future research of the speed of growth of order of our base is important. On the other hand DLP for cyclic subgroups generated by a multivariate map is known as a difficult problem . We use computer simulation for the studies of our polynomial maps densities in the case of Boolean rings $B(m)$ and compare these results with similar simulations in the case of fields $\mathbb{F}_2^m$. So the reader can judge the feasibility of our symbolic algorithms as tools for access control, digital signatures, generation of hash functions.

Table 1: Public map generation time (ms), $D(n, \mathbb{B}(32))$, case I

	length of the word			
n	32	64	128	256
16	16	24	40	72
32	112	232	476	952
64	2104	4332	8960	18385
128	32229	66591	136928	284461

[Generation of Graph Based Multivariate Maps]

We use the term *density* for the number of monomial expressions of multivariate function.

We use computer simulation to generate maps of kind $\mathrm{y} = \tau_1 N_{\alpha_1} N_{\alpha_2} \dots N_{\alpha_s} \tau_2(\mathrm{x})$ related to graphs $A(n, K)$ and $D(n, K)$. K is one of the commutative rings: Boolean ring $\mathbb{B}(32)$, modular ring $\mathbb{Z}_{2^{32}}$ and finite field $\mathbb{F}_{2^{32}}$.

We have implemented three cases of invertible affine transformations:

1. τ_1 and τ_2 are identities,
2. τ_1 and τ_2 are of kind $x_1 \to x_1 + a + 2x_2 + a_3 x_3 + \dots + a_{n+1} xn + 1$ (linear time of computing τ_1 and τ_2),
3. $\tau_1 = A_1 x + b_1$ and $\tau_2 = A_2 x + b_2$; matrices A_1, A_2 and vectors b_1, b_2 has mostly nonzero elements.

Tables 1–18 and figures 1–6 presents the density and the generation time of the maps.

Tables 19–20 presents the time of encryption with symmetric algorithm and three different commutative rings.

References

[1] B. Bollobás, *Extremal Graph Theory.* Extremal Graph Theory, 1978.

[2] R. E. Chen, Ed., *Cryptography Research Perspectives.* USA: Nova Publishers, 2009.

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

	length of the word			
n	32	64	128	256
16	20	28	44	80
32	112	232	472	952
64	1964	4172	8604	17457
128	29929	64439	137824	285473

Table 3: Public map generation time (ms), $D(n, \mathbb{F}_{2^{32}})$, case I

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

Table 4: Public map generation time (ms), $A(n, \mathbb{B}(32))$, case I

	length of the word			
n	32	64	128	256
16	12	20	36	68
32	116	228	460	952
64	1825	3805	7785	15814
128	29886	62988	133209	281298

Table 5: Public map generation time (ms), $A(n, \mathbb{Z}_{2^{32}})$, case I

	length of the word			
n	32	64	128	256
16	12	20	36	76
32	116	244	500	1008
64	1880	3972	8441	17549
128	29971	64709	140510	302148

Table 6: Public map generation time (ms), $A(n, \mathbb{F}_{2^{32}})$, case I

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

Table 7: Public map generation time (ms), $D(n, \mathbb{B}(32))$, case II

	length of the word			
n	32	64	128	256
16	20	36	60	108
32	164	336	676	1352
64	2660	5480	11305	23502
128	82304	175455	362382	751748

Table 8: Public map generation time (ms), $D(n, \mathbb{Z}_{2^{32}})$, case II

	length of the word			
n	32	64	128	256
16	16	28	56	104
32	168	344	700	1428
64	2856	6112	12620	25652
128	80227	179877	398918	842802

Table 9: Public map generation time (ms), $D(n, \mathbb{F}_{2^{32}})$, case II

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

[3] G. A. Margulis, "Explicit group-theoretical constructions of combinatorial schemes and their application to the design of expanders and concentrators," *Probl. Inf. Transm.*, vol. 24, no. 1, pp. 39–46, 1988.

[4] P. S. Guinand and J. H. Lodge, "Tanner type codes arising from large girth graphs," in *Proceedings of the 1997 Canadian Workshop on Information Theory (CWIT '97)*, Toronto, 1997, pp. 5–7.

Table 10: Public map generation time (ms), $A(n, \mathbb{B}(32))$, case II

	length of the word			
n	32	64	128	256
16	16	28	52	100
32	144	304	632	1304
64	2489	5113	10401	21386
128	78744	162213	333259	678501

Table 11: Public map generation time (ms), $A(n, \mathbb{Z}_{2^{32}})$, case II

	length of the word			
n	32	64	128	256
16	4	16	44	92
32	172	348	708	1428
64	2617	5601	11962	24707
128	75669	162459	350227	761242

Table 12: Public map generation time (ms), $A(n, \mathbb{F}_{2^{32}})$, case II

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

Table 13: Public map generation time (ms), $D(n, \mathbb{B}(32))$, case III

	length of the word			
n	32	64	128	256
16	16	32	56	108
32	240	416	764	1464
64	5357	8509	14802	27391
128	192324	310666	547293	1020502

Table 14: Public map generation time (ms), $D(n, \mathbb{Z}_{2^{32}})$, case III

	length of the word			
n	32	64	128	256
16	20	32	56	104
32	260	440	800	1520
64	5524	8780	15180	28381
128	180436	289475	507985	945409

Table 15: Public map generation time (ms), $D(n, \mathbb{F}_{2^{32}})$, case III

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

Table 16: Public map generation time (ms), $A(n, \mathbb{B}(32))$, case III

	length of the word			
n	32	64	128	256
16	20	24	52	92
32	244	412	760	1452
64	5400	8628	15073	27961
128	192871	312039	550666	1027075

Table 17: Public map generation time (ms), $A(n, \mathbb{Z}_{2^{32}})$, case III

	length of the word			
n	32	64	128	256
16	24	36	64	116
32	248	428	788	1508
64	5317	8576	15216	28176
128	180861	290432	509812	949652

[5] P. S. Guinand and J. H. Lodge, "Graph theoretic construction of generalized product codes," in *Proceedings of IEEE International Symposium on Information Theory*, 1997, p. 111.

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

[7] N. Biggs, *Algebraic Graph Theory.* Cambridge University Press, 1993.

[8] N. Biggs, "Graphs with large girth," *Ars Combi-*

Table 18: Public map generation time (ms), $A(n, \mathbb{F}_{2^{32}})$, case III

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

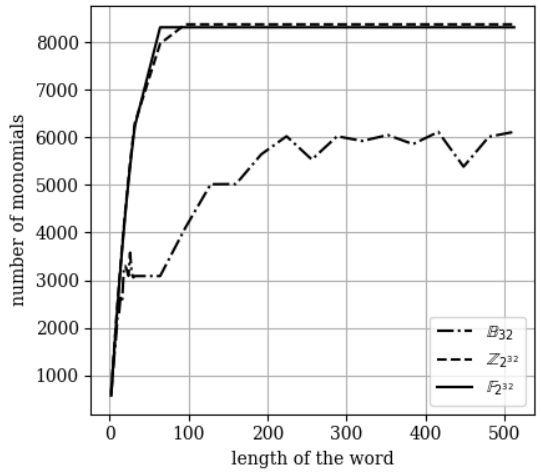

Figure 1: Number of monomials in public map ($n = 128$), graph $D(n, K)$, $K = \mathbb{B}(32), \mathbb{Z}_{2^{32}}, \mathbb{F}_{2^{32}}$, case I

Table 19: Encryption time (ms) for fixed password length (64B), graph $D(n, K)$, $K = \mathbb{B}(32), \mathbb{Z}_{2^{32}}, \mathbb{F}_{2^{32}}$

file size	$\mathbb{B}(32)$	$\mathbb{Z}_{2^{32}}$	$\mathbb{F}_{2^{32}}$
4MB	48	52	1816
8MB	88	104	3564
16MB	180	212	7344
32MB	357	400	14697

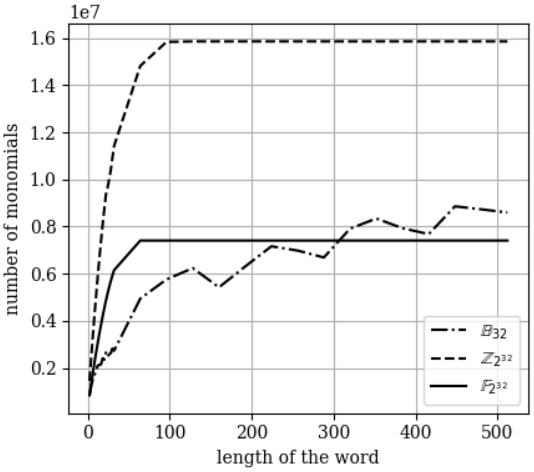

Figure 2: Number of monomials in public map ($n = 128$), graph $D(n, K)$, $K = \mathbb{B}(32), \mathbb{Z}_{2^{32}}, \mathbb{F}_{2^{32}}$, case II

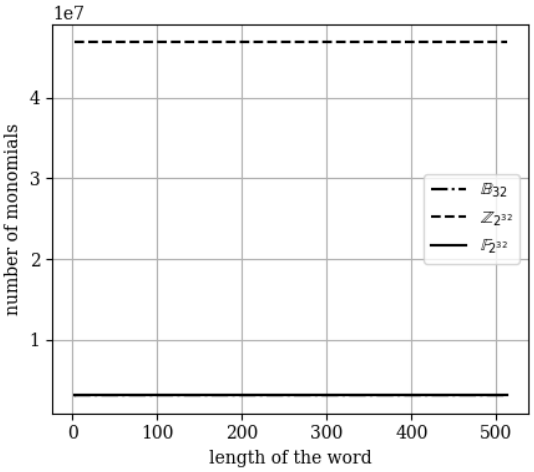

Figure 3: Number of monomials in public map ($n = 128$), graph $D(n, K)$, $K = \mathbb{B}(32), \mathbb{Z}_{2^{32}}, \mathbb{F}_{2^{32}}$, case III

natoria, vol. 25C, pp. 73–80, 1988.

[9] Jon-Lark Kim, U. N. Peled, I. Perepelitsa, V. Pless, and S. Friedland, "Explicit construction of families of ldpc codes with no 4-cycles," *IEEE T. Inform. Theory*, vol. 50, no. 10, pp. 2378–2388, 2004.

[10] A. Lubotzky, R. Phillips, and P. Sarnak, "Ramanujan graphs," *Combinatorica*, vol. 8, no. 3, pp. 261–277, 1988.

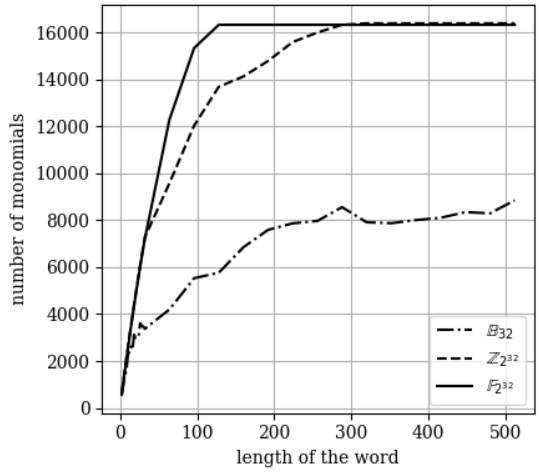

Figure 4: Number of monomials in public map ($n = 128$), graph $A(n, K)$, $K = \mathbb{B}(32), \mathbb{Z}_{2^{32}}, \mathbb{F}_{2^{32}}$, case I

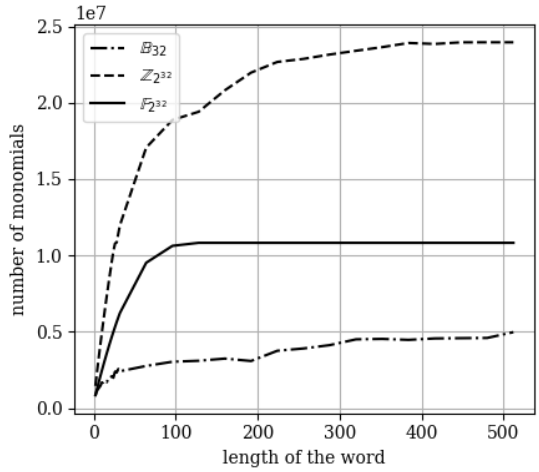

Figure 5: Number of monomials in public map ($n = 128$), graph $A(n, K)$, $K = \mathbb{B}(32), \mathbb{Z}_{2^{32}}, \mathbb{F}_{2^{32}}$, case II

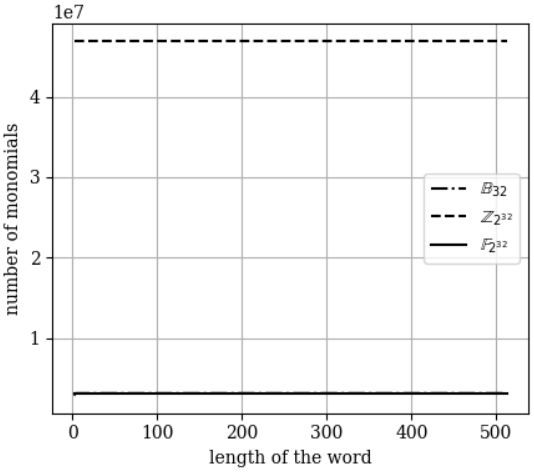

Figure 6: Number of monomials in public map ($n = 128$), graph $A(n, K)$, $K = \mathbb{B}(32), \mathbb{Z}_{2^{32}}, \mathbb{F}_{2^{32}}$, case III

Table 20: Encryption time (ms) for fixed password length (64B), graph $A(n, K)$, $K = \mathbb{B}(32), \mathbb{Z}_{2^{32}}, \mathbb{F}_{2^{32}}$

file size	$\mathbb{B}(32)$	$\mathbb{Z}_{2^{32}}$	$\mathbb{F}_{2^{32}}$
4MB	40	44	1712
8MB	76	96	3524
16MB	152	192	7012
32MB	300	329	13556

vol. 1, pp. 215–235, 1966.

[13] N. Koblitz, *A Course in Number Theory and Cryptography*. Berlin, Heidelberg: Springer-Verlag, 1987.

[14] A. Cossidente and M. J. de Resmini, "Remarks on singer cyclic groups and their normalizers," *Design Code Cryptogr.*, vol. 32, pp. 97–102, 2004.

[15] W. M. Kantor, "Linear groups containing a singer cycle," *J. Algebra*, vol. 62, no. 1, pp. 232–234, 1980.

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

[12] P. Erdös, A. Rényi, and V. T. Sós, "On a problem of graph theory," *Stud. Sci. Math. Hungar.*,