Proceedings of the Federated Conference on Computer Science and Information Systems pp. 397–405
DOI: 10.15439/2018F204
ISSN 2300-5963 ACSIS, Vol. 15

On the implementation of new symmetric ciphers based on non-bijective multivariate maps

Vasyl Ustymenko, Aneta Wróblewska
Maria Curie-Skłodowska University,
Institute of Mathematics,
pl. Marii Curie-Skłodowskiej 1,
20-031 Lublin, Poland
Email: vasylustimenko@yahoo.pl
awroblewska@hektor.umcs.lublin.pl

Urszula Romańczuk-Polubiec
Independent Researcher, Poland
Email: urszula_romanczuk@yahoo.pl

Monika Polak
Rochester Institute of Technology,
Department of Computer Science,
20 Lomb Memorial Dr,
Rochester, NY 14623, USA
Email: mkp@cs.rit.edu

Eustrat Zhupa
University of Rochester,
Department of Computer Science,
Rochester, NY 14627, USA
Email: ezhupa@cs.rochester.edu

***Abstract*—Certain families of graphs can be used to obtain multivariate polynomials for cryptographic algorithms. In particular, in this paper, we introduce stream ciphers based on non-bijective multivariate maps. The presented symmetric encryption algorithms are based on three families of bipartite graphs with partition sets isomorphic to $\mathbb{K}^n$, where $\mathbb{K}$ is selected as the finite commutative ring. The plainspace of the algorithm is $\Omega = \{x | \sum x_i \in \mathbb{K}^*, x \in \mathbb{K}^n\} \subset \mathbb{K}^n$, $\Omega \cong \mathbb{K}^* \times \mathbb{K}^{n-1}$. We describe the algorithm for the case $\mathbb{K} = \mathbb{Z}_{2^m}$, $m \geq 2$. In fact, we use the relation $d * d_{dec} \equiv 1 (\mathbf{mod}\ 2^{m-1})$, $d, d_{dec} \in \mathbb{Z}^*_{2^{m-1}}$ to obtain encryption polynomial map of degree greater than or equal to $d+2$ and decryption map of degree greater than or equal to $d_{dec}+2$. We assume d_{dec} grows with the growth of parameter m, because this makes cryptanalysis very difficult task. Symmetric encryption and decryption algorithms for users are numerical recurrent processes, not requiring generation of encryption and decryption maps in their symbolic forms. They use arithmetical operations of addition, subtraction, and multiplication. That's why the algorithms are robust (execution speed is $O(n)$). To break the algorithm an adversary must use linearization attacks for recovering non-bijective "decryption map" of degree greater than $d_{\mathrm{dec}}+2$ in its symbolic form. To achieve this, the adversary needs at least $O(n^{d_{\mathrm{dec}}}+2)$ pairs of plaintext and corresponding ciphertext to restore the non-bijective map of degree greater than or equal to $d_{dec}+2$. We present tables for evaluation of execution time for m = 8 with various length of passwords and sizes of files. Computer simulations demonstrate good mixing properties of the encryption functions.**

FEW graph based algorithms have been implemented since 1998 (see [1] - [25]). So there is some history of the usage of sparse algebraic graphs in symmetric cryptographical algorithms. The following known graphs defined over finite commutative ring $\mathbb{K}$ were used: $D(n, \mathbb{K})$ (see [1], for $\mathbb{K} = \mathbb{F}_q$ graphs were defined and investigated in [26], [27]), $W(n, \mathbb{K})$ (Wenger graphs defined in [28]), graphs $A(n, \mathbb{K})$ introduced in [45] and graphs $\widetilde{D(n, \mathbb{K})}$ of [25]. Popular choices of $\mathbb{K}$ are finite fields $\mathbb{F}_{127}$, $\mathbb{F}_{2^7}$, $\mathbb{F}_{2^8}$, $\mathbb{F}_{2^{16}}$ and $\mathbb{F}_2^{32}$ and rings modular arithmetics $\mathbb{Z}_{2^7}$, $\mathbb{Z}_{2^8}$, $\mathbb{Z}_{2^{16}}$. We present this research history in the next section.

In section 3 we introduce a class of bivariate graphs containg all the above mentioned graphs. Such concept is convenient for uniform description of encryption scheme and observation of common properties of graphs from this class (section 4). We compare graphs and related algorithms corresponding to different families ($W(n, \mathbb{K})$, $D(n, \mathbb{K})$, $A(n, \mathbb{K})$ and $\widetilde{D(n, \mathbb{K})}$) in section 5.

Here the reader can find remarks on multivariate cryptography and its connections with cryptographical applications of Algebraic Graph Theory.

RSA is one of the most popular cryptosystems. It is based on a number factorization problem and on Euler's Theorem. Peter Shor discovered that factorization problem can be effectively solved by using a theoretical quantum computer. It means that RSA could not be a security tool in the future postquantum era. One of the research directions leading to a postquantum secure public key is the Multivariate Cryptography which uses a polynomial maps of affine space $\mathbb{K}^n$ defined over a finite commutative ring $\mathbb{K}$ into itself as encryption tools (see [29]). This is a young promising research area because of the current lack of known cryptosystems with the proven resistance against attacks with the use of Turing machines. Other important direction of Postquantum Cryptography is the study of Hyperelliptic Curves Cryptosystems. We have to say that classical elliptic curves encryption will be not secure in the Postquantum era.

Applications of Algebraic Graph Theory to Multivariate Cryptography were shown in our talks at Erdös Centennial (2013, Budapest) and Central European Conference on Cryptology 2014 (Alfred Renyi Institute, Budapest) [30], [31]. Talks were devoted to algorithms based on bijective maps of affine spaces into itself. Applications of algebraic graphs to cryptography started with symmetric algorithms based on explicit constructions of extremal graph theory and their directed analogues (see survey [11], [32]). The main idea is to convert an algebraic graph in a finite automaton and to use the pseudorandom walks on the graph as encryption tools.

This approach can also be used for the key exchange protocols. Nowadays the idea of "symbolic walks" on algebraic graphs, when the walk on the graph depends on parameters given as special multivariate polynomials in variables depending from plainspace vector, appears in several public key cryptosystems. Another source of graphs suitable for cryptography is connected to finite geometries and their flag system (see [33] and further references).

Multivariate cryptography started from the study of potential for the special quadratic encryption multivariate bijective map of $\mathbb{K}^n$, where $\mathbb{K}$ is an extention of finite field $\mathbb{F}_q$ of characteristic 2. One of the first such cryptosystems was proposed by Imai and Matsumoto and cryptanalysis for that system was invented by J. Patarin. A survey on various modifications of this algorithm and corresponding cryptanalysis can be found in [29] or [34].

One of the first uses of non-bijective map of multivariate cryptography was in the *oil and vinegar* cryptosystem proposed in [35] and analyzed in [36]. Nowadays, this general idea is strongly supported by publication [37] devoted to security analysis of direct attacks on modified unbalanced oil and vinegar systems. It looks like such systems and rainbow signature schemes may lead to promising Public Key Schemes of Multivariate Encryption defined over finite fields. Non-bijective multivariate sparse encryption maps of degree 3 and ≥ 3 based on walks on algebraic graphs $D(n, \mathbb{K})$ defined over general commutative ring and their homomorphic images were proposed in [38]. Security of the corresponding cryptosystem rests on the idea of hidden discrete logarithm problem. U. Romańczuk-Polubiec and V. Ustimenko combine an idea of "oil and vinegar signature cryptosystem" with the idea of linguistic graph-based map with partially invertible decomposition to introduce a new cryptosystem [38]. This algorithm can be implemented with the use of families $D(n, \mathbb{K})$ and $A(n, \mathbb{K})$ and natural homomorphism between them. Finally, in [39] "hidden RSA multivariate encryption" based on graphs $D(n, \mathbb{K})$ were proposed.

In this paper we modify the encryption map (private key) of the above mentioned cryptosystem in terms of family of bivariate graphs defined over the commutative ring $\mathbb{K}$. These maps have multivariate nature despite the "numerical implementation" in symmetric ciphers mode with the plainspace isomorphic to $\mathbb{K}^* \times \mathbb{K}^{n-1}$.

I. On some implementation of algorithms based on bijective maps

We worked on a software package that allows us to investigate strongly symmetric cases of stream ciphers based on graphs $W(n, \mathbb{K})$, $D(n, \mathbb{K})$, $\widetilde{D(n, \mathbb{K})}$ and $A(n, \mathbb{K})$, where $\mathbb{K}$ is the arithmetic ring. Some cases are already implemented by our team at the level of prototype model.

In very special cases the algorithms were previously implemented. The first implementation of $D(n, \mathbb{K})$ encryption was done in 2000 at the University of South Pacific (USP, Fiji Islands). The research team was composed by Prof. V. Ustimenko, PhD Dharmendra Sharma (currently professor of University of Canberra), postgraduate students V. Gounder and R. Prasad (see [2], [3]). The work was supported by the University Research Committee of the University of South Pacific (USP) grant. The implementation of this case on asymmetric mode was discussed in [5]. The chosen case for $\mathbb{K}$ was $\mathbb{F}_{127}$, which is the closest prime number to the size of ASCII code alphabet. It means that one has to delete just the *delete* service symbol and can encrypt arbitrary files of type txt. The chosen string was $\alpha_i(x) = x + d_i$, where d_i are elements of chosen ring $\mathbb{K} = \mathbb{F}_{127}$ chosen in pseudorandom fashion. So that was a case of shifting encryption.

The affine transformations L_1 and L_2 were simply identities. Implemented cipher on ordinary PC was rather robust in performance, but with average mixing properties. It's been used at USP digital network working for campuses and USP centers located in 11 island countries of South Pacific region. The package was also used by ORACLE based system of the bursary office (see [8]). Recently group of students from Okanagan college (affiliated with the University of British Columbia) implemented that stream cipher on a cluster network of PC's. It was used for a large data encryption [10]. The implementation of that security algorithm for protection of Geo Information Systems was described in [6], [7].

Another case for $\mathbb{K} = \mathbb{Z}_{256}$ and graph $D(n, \mathbb{K})$ was implemented under the Research Committee of Sultan Qaboos University (SQU, Oman) grant. The research team was composed of professors Vasyl Ustimenko and Abderezak Tousane and students Rahma Al Habsi and Huda Al Naamani. The software uses one to one correspondence between element of $\mathbb{Z}_{256}$ and symbols of binary alphabet. It allows encryption of various file types (with extension doc, jpg, htm, avi, pdf, ...) in a way that encrypted file is presented in the same format with the plaintext. The symmetric algorithm was used at academical networks of SQU and Kiev Mohyla Academy [9], [10].

The cases of $D(n, \mathbb{K})$, where $\mathbb{K}$ is the finite field $\mathbb{F}_{2^7}$ $\mathbb{F}_{2^8}$, $\mathbb{F}_{2^{16}}$, the shifting encryption was implemented and investigated in [20].

The systematic study of shifting encryption for cases of shifting encryptions of $D(n, \mathbb{K})$ was conducted at UMCS (Lublin, Poland). J. Kotorowicz used arithmetical rings $\mathbb{Z}_2^7$, $\mathbb{Z}_2^8$, $\mathbb{Z}_2^{16}$ for the implementation with various affine transformation τ_L and τ_R (see [14], [16]). The encryption was essentially faster than in all previously known cases. The selected affine transformation leads to an encryption with very good mixing properties: the change of a single character of the plaintext or the change of a single character of the encryption string $d_1, d_2, \ldots, d_s$ causes the change of at least 98 percent of the ciphertext characters. In [23] these cases were implemented for graphs $A(n, \mathbb{K})$ with very similar results on the mixing properties. In the case of $\tau_R = {\tau_L}^{-1}$ it can be proved that the order of $A(n, \mathbb{K})$ and $D(n, \mathbb{K})$ based encryption map grows with the growth of parameter n. The comparison of orders was completed through the study of cycles structures of $A(n, \mathbb{K})$ and $D(n, \mathbb{K})$ encryptions. Results demonstrated similarity in both cases.

M. Klisowski implemented $D(n,\mathbb{K})$ and $A(n,\mathbb{K})$ shifting encryption on symbolic level in the cases of finite fields $\mathbb{F}_{2^7}$ $\mathbb{F}_{2^8}$, $\mathbb{F}_{2^{16}}$ $\mathbb{F}_{2^{32}}$ ([21], [22], [24]). In [40] A. Wróblewska proved that shifting $D(n,\mathbb{K})$ encryption is given by a cubical multivariate map. A similar result for $A(n,\mathbb{K})$ based encryption was stated in [41]. Simulation results of [22], [23] allow to estimate time of generation of these maps as functions of parameter n and densities of such multivariate cubic encryption and decryption maps. A comparison of cases $A(n,\mathbb{K})$ and $D(n,\mathbb{K})$ for the above fields an be found in [24]. Similar results for cases of Boolean rings of sizes 2^7, 2^8, 2^{16}, 2^{32} are obtained via computer simulations.

The PhD Thesis of M. Klisowski [42] contains the first results on $D(n,\mathbb{K})$ and $A(n,\mathbb{K})$ based multivariate maps which are not defined via shifting encryptions. He used symbolic strings of kind $\alpha_1(x) = x+c_1, \alpha_2(x) = x+c_2, \ldots, \alpha_{s-1}(x) = x + c_{s-1}$, $\alpha_s(x) = x^3 + c_s$ with constants c_i, $i = 1, 2, \ldots, s$ for special fields $\mathbb{F}_q$ in which $x^3 = b$ has unique solution. It was shown that such a choice makes direct linearization attacks impossible.

The first implementation for the case of Wenger graph based encryption was completed at the University of Sao Paolo (USP, Brasil) (see [12] and further references). Professors V. Futorny and V. Ustimenko chose field $\mathbb{F}_{253}$ which size is the closest from below prime to the size of binary alphabet. This research was partially supported by FAPESP foundation (grant for international cooperation with USP). Computer simulation demonstrated high speed of encryption. In [12] authors evaluated the diameter of graph $W(n,\mathbb{F}_q)$ and proved that the family of these graphs $W(n,q)$, $n \leq q$ is a family of small world graphs.

Professor Routo Terada (USP, Brasil) suggested to investigate the behaviour of these algorithms under linearization attacks. Computer simulation supports the conjecture on a good resistance of the encryption scheme to such attacks.

The idea of using graphs $A(n,\mathbb{K})$ in cryptography was proposed by U. Romańczuk-Polubiec and V. Ustimenko in [45]. Evaluation of the order of encryption map based on $A(n,q)$ was presented in [23]. A theoretical study of orders and cycles can be found in [44], [45].

Some stream ciphers defined via graphs $\widetilde{D(n,\mathbb{K})}$ were proposed by M. Polak and V. Ustymenko in [25]. Furthermore, M. Polak compared LDPC codes corresponding to $A(n,\mathbb{K})$, $D(n,\mathbb{K})$ and $\widetilde{D(n,\mathbb{K})}$ in [49].

The importance of such graphs was justified in [44]. The encryption algorithm was implemented and some properties (speed, mixing properties, order) were investigated in the paper.

II. On the class of bivariate graphs

Let $\mathbb{K}$ be a commutative ring. We define $T(n,\mathbb{K})$ as a bipartite graph with the set of vertices $V(T) = P \cup L$, $P \cap L = \emptyset$. We call $P = \mathbb{K}^n$ a set of points and $L = \mathbb{K}^n$ a set of lines (two copies of a Cartesian power of $\mathbb{K}$ are used). We will use two types of brackets to distinguish points $(p) \in P$ and lines $[l] \in L$:

$$(p) = (p_1, p_2, \ldots, p_n) \in P,$$

$$[l] = [l_1, l_2, \ldots, l_n] \in L.$$

p_i, l_i $(1 \leq i \leq n)$ are elements of $\mathbb{K}$. We say that vertex (p) (point (p)) is incident with the vertex $[l]$ (line $[l]$) and we write: $(p)I_T[l]$, if the following relations between their coordinates hold:

$$\begin{cases} p_2 - l_2 = e_2^1 p_1 l_1 \\ p_3 - l_3 = e_3^1 p_1 l_2 + e_3^2 l_1 p_2 \\ \vdots \\ p_s - l_s = e_s^1 p_1 l_{i_s} + e_s^2 l_1 p_{j_s} \\ \vdots \\ p_n - l_n = e_n^1 p_1 l_{i_n} + e_n^2 l_1 p_{j_n} \end{cases} \quad (1)$$

where $e_2^1, e_s^1, e_s^2 \in \{0, 1, -1\}$, $1 \leq i_s < s$, $1 \leq j_s < s$. So the incidence relations for graph $T = T(n,\mathbb{K})$ are given by condition $(p)I_T[l]$. The set of edges consists of all pairs $\{(p), [l]\}$ for which: $(p)I_T[l]$. Let us consider the case of finite commutative ring $\mathbb{K}$, $|\mathbb{K}| = k$. As it instantly follows from the definition, the order of our bipartite graph is $|V(T)| = 2k^n$ and the number of edges is $|E(T)| = k^n \cdot k = k^{n+1}$. Graphs $T = T(n,\mathbb{K})$ are k-regular. In fact, the neighbour of a given point (p) is given by above equations, where parameters $p_1, p_2, \ldots, p_n$ are fixed elements of the ring and symbols $l_1, l_2, \ldots, l_n$ are variables. It is easy to see that if we set l_1 then this choice uniformly establishes values $l_2, l_3 \ldots, l_n$. So each point has precisely k neighbours. In a similar way we observe that the neighbourhood of any line also contains k neighbours. Notice, that the order and degree of our graph defined via strings i_s, j_s, e_2^1, e_s^1, e_s^2, where $s = 2, 3, \ldots, n$, does not depend on the strings.

Let us consider some examples.

Wenger graphs $W(n,\mathbb{K})$

In 1991 Wenger defined the family of bipartite, p-regular graphs $H_n(p)$, where p prime number [28]. In [26] Lazebnik and Ustimenko introduced straight forward generalization $W(n,q)$ of these graphs via change of $\mathbb{F}_p$ to $\mathbb{F}_q$, where q is a prime power. They used special Lie algebra and proved that the family of bipartite, q-regular graphs $W(n,q)$, where q is prime power and $n \geq 2$. Graphs $W(n,q)$ are defined for all prime powers and $H_n(p) = W(n,p)$ are defined only for primes.

The set of vertices of infinite incidence structure (P, L, I) is $V = P \cup L$ and the set of edges E consists of all pairs $\{(p), [l]\}$ for which $(p)I[l]$. Bipartite graphs $W(n,q)$ have partition sets P_n (collection of points) and L_n (collection of lines) isomorphic to vector space $\mathbb{F}_q^n$, where $n \in \mathbb{N}_+$. Let us use the following notations for points and lines in graph $W(n,q)$:

$$(p) = (p_1, p_2, p_3, \ldots, p_n) \in P,$$

$$[l] = [l_{1,}, l_2, l_3, \ldots, l_n] \in L.$$

The point (p) is incident with the line $[l]$, and we write $(p)I_W[l]$, if the following relations between their coordinates hold:

$$\{\ l_i - p_i = p_1 l_{i-1}, \tag{2}$$

for $2 \leq i \leq n$. The graphs $W(n, \mathbb{F}_q)$ have cycles of length 8.

One can change finite field $\mathbb{K}$ for general commutative ring $\mathbb{K}$ and work with graph $W(n, \mathbb{K})$.

Graphs $A(n, \mathbb{K})$

Graphs $A(n, \mathbb{K})$, formally appearing as graphs $E(n, \mathbb{K})$ in [43], are used as tools for the study of $D(n, \mathbb{K})$ properties. Later on the graphs $E(n, \mathbb{K})$ were presented with another name as an independent family $A(n, q)$ for the first time in [45] for cryptographic applications.

Let us use the following notations for points and lines in the graph $A(n, \mathbb{K})$:

$$(p) = (p_1, p_2, p_3, \ldots, p_n) \in P,$$
$$[l] = [l_{1,}, l_2, l_3, \ldots, l_n] \in L.$$

The point (p) is incident with the line $[l]$, and we write $(p)I_A[l]$, if the following relations between their coordinates hold:

$$\begin{cases} l_2 - p_2 = l_1 p_1 \\ l_3 - p_3 = p_1 l_2 \\ l_4 - p_4 = l_1 p_3 \\ l_i - p_i = p_1 l_{i-1} \text{ for odd } i \\ l_i - p_i = l_1 p_{i-1} \text{ for even } i \end{cases} \tag{3}$$

for $3 \leq i \leq n$.

Graphs $D(n, \mathbb{K})$

The following interpretation of a family of graphs $D(n, \mathbb{K})$ in case $\mathbb{K} = \mathbb{F}_q$ can be found in [27]. By I_D we denote the incidence relation for this graph. Let us use the following notations for points and lines:

$$(p) = (p_1, p_2, p_3, \ldots, p_n) \in P,$$
$$[l] = [l_{1,}, l_2, l_3, \ldots, l_n] \in L.$$

Two types of brackets allow us to distinguish points from lines. Points and lines are elements of two copies of the vector space over $\mathbb{K}$. Point (p) is incident with the line $[l]$, and we write $(p)I_D[l]$, if the following relations between their coordinates hold:

$$\begin{cases} l_2 - p_2 = l_1 p_1 \\ l_3 - p_3 = p_1 l_2 \\ l_4 - p_4 = l_1 p_2 \\ l_i - p_i = p_1 l_{i-2} \text{ for } i \bmod 4 \equiv 2 \text{ or } i \bmod 4 \equiv 3 \\ l_i - p_i = l_1 p_{i-2} \text{ for } i \bmod 4 \equiv 0 \text{ or } i \bmod 4 \equiv 1 \end{cases} \tag{4}$$

where $3 \leq i \leq n$.

The set of vertices is $V = P \cup L$ and the set of edges E consists of all pairs $\{(p), [l]\}$ for which $(p)I_D[l]$. Bipartite graphs $D(n, \mathbb{K})$ have partition sets P (collection of points) and L (collection of lines) isomorphic to vector space $\mathbb{K}^n$, where $n \in \mathbb{N}_+$.

Graphs $\widetilde{D(n, \mathbb{K})}$

Formal definitions for the family of graphs $\widetilde{D(n, \mathbb{K})}$ were presented in [25].

Construction of projective limits graphs of $\widetilde{D(n, \mathbb{K})}$ appears in papers motivated by results on embeddings of Chevalley group geometries in the corresponding Lie algebras and construction of blow-up for an incidence system of Weyl groups in [46], [47]. Moreover, this structure is the base for construction of family of graphs $D(n, \mathbb{K})$ (see [25, 27]).

Let us use the analogical notations for points and lines in graph $\widetilde{D(K)}$:

$$(p) = (p_1, p_2, p_3, \ldots, p_n) \in P,$$
$$[l] = [l_{1,}, l_2, l_3, \ldots, l_n] \in L.$$

In the incidence structure $\widetilde{(P, L, I)}$ the point (p) is incident with the line $[l]$, and we write $(p)I_{\widetilde{D}}[l]$, if the following relations between their coordinates hold:

$$\begin{cases} l_2 - p_2 = l_1 p_1 \\ l_3 - p_3 = p_1 l_2 \\ l_4 - p_4 = l_1 p_2 \\ l_5 - p_5 = l_1 p_3 - p_1 l_4 \\ l_i - p_i = p_1 l_{i-1} \text{ for } i \bmod 3 \equiv 0 \\ l_i - p_i = l_1 p_{i-2} \text{ for } i \bmod 3 \equiv 1 \\ l_i - p_i = l_1 p_{i-2} - p_1 l_{i-1} \text{ for } i \bmod 3 \equiv 2 \end{cases} \tag{5}$$

for $3 \leq i \leq n$.

Graphs from families $D(n, \mathbb{K})$ and $\widetilde{D(n, \mathbb{K})}$ are bipartite, k-regular, where $|\mathbb{K}| = k$. The girth of graphs from the described families increases with the growth of n. In fact $D(n, q)$ is a family of graphs of large girth and there is a conjecture that $\widetilde{D(n, q)}$ is another family of graphs of a large girth.

All graphs from the considered families are k-regular, bipartite and the set of vertices is $V = P \cup L$, $P \cap L = \emptyset$. They are sparse graphs.

It is clear that there is a natural homomorphism of $T(n+1, \mathbb{K})$ onto $T(n, \mathbb{K})$ of "deleting the last coordinate" that sends $(x_1, x_2, \ldots, x_n, x_{n+1})$ to $(x_1, x_2, \ldots, x_n)$ and $[y_1, y_2, \ldots, y_n, y_{n+1}]$ to $[y_1, y_2, \ldots, y_n]$. It means that there is a well defined projective limit $T(K)$ of graphs $T(n, \mathbb{K})$, $n \to \infty$. Bivariate graphs form a special subclass of so called *linguistic graphs* for which natural projective limits are defined in a similar way.

Recall that the girth $g = g(\Gamma)$ of the graph Γ is the length of its minimal cycle.

Let us assume that the girth $g(n)$ of graphs $T(n, \mathbb{K})$ is unbounded. The obvious inequality $g(n+1) \geq g(n)$ holds. It means that projective limit $T(\mathbb{K})$ has to be a $|\mathbb{K}|$-regular forest. We have such situation in cases of graphs $A(n, \mathbb{F}_q)$ and $D(n, \mathbb{F}_q)$ If $q \geq 2$ then $A(\mathbb{F}_q)$ is a single tree presented by the above equations. Graph $D(F_q)$ is an infinite forest containing infinitely many trees.

Projective limit $W(\mathbb{F}_q)$ of Wenger graphs is an infinite connected graph containing cycles of length 8.

III. General encryption algorithm

We can convert graph $T(n,\mathbb{K})$ to finite automaton in the following way. Let $\mathrm{v}=(v_1,v_2,v_3,v_4,\dots,v_n)\in V(T(n,\mathbb{K}))$ (or $\mathrm{v}=[v_1,v_2,v_3,v_4,\dots,v_n]\in V(T(n,\mathbb{K}))$) and $N_\alpha(\mathrm{v})$ be the operator of taking neighbor of vertex v where the first coordinate is α:

$$N_\alpha(v_1,v_2,v_3,v_4,\dots,v_n)\to[\alpha,*,*,*,\dots,*],$$
$$N_\alpha[v_1,v_2,v_3,v_4,\dots,v_n]\to(\alpha,*,*,*,\dots,*),$$

where $\alpha\in\mathbb{K}$. The remaining coordinates can be determined uniquely using relations describing the chosen graph $T(n,\mathbb{K})$.

We convert $T(n,\mathbb{K})$ to finite automaton via joining v an $N_\alpha(\mathrm{v})$ by directed arrow with weight α. We assume that all vertices of the graph are accepting states.

A bit more interesting object is a symbolic bivariate automaton. Let $\mathrm{a}(x)=(\alpha_1(x),\alpha_2(x),\dots,\alpha_s(x))$ be a string of elements from $\mathbb{K}[x]$ (totality of polynomials in variable x with coefficients from K).

We introduce operator $N^s_{\mathrm{a(x)}}(\mathrm{v})$, where v is a point or a line with coordinates $v_1,v_2,\dots,v_n$, of taking the last vertex u of the path v, $\mathrm{v}_1=N_{\alpha_1(v_1)}(\mathrm{v})$, $\mathrm{v}_2=N_{\alpha_2(v_1)}(\mathrm{v}_1)$, …, $\mathrm{v}_s=N_{\alpha_s(v_1)}(\mathrm{v}_{s-1})=\mathrm{u}$.

We refer to $N^s_{\mathrm{a(x)}}$ as a computation of the symbolic automaton with the string

$$\mathrm{a(x)}=(\alpha_1(\mathrm{x}),\alpha_2(\mathrm{x}),\dots,\alpha_\mathrm{s}(\mathrm{x}))$$

$\alpha_i\in\mathbb{K}[x]$, $i=1,\dots,s$ and initial state $\mathrm{v}=(v_1,v_2,v_3,v_4,\dots,v_n)\in T(n,\mathbb{K})$ (or $\mathrm{v}=[v_1,v_2,v_3,v_4,\dots,v_n]\in T(n,\mathbb{K})$). We can consider $F_s(\mathrm{v})=N^s_{\mathrm{a(v_1)}}(\mathrm{v})$ as a map on $P\cup L$.

It is easy to see that the restriction of this map on P is a polynomial transformation of $P=\mathbb{K}^n$ into P (parameter s is even) or L (parameter s is odd) of kind

$$\begin{aligned}x_1&\to f_1(x_1,x_2,\dots,x_n),\\ x_2&\to f_2(x_1,x_2,\dots,x_n),\\ &\vdots\\ x_n&\to f_n(x_1,x_2,\dots,x_n).\end{aligned}$$

Notice that generally F_s is not a bijection. Let us consider an invertibility condition for F_s.

Proposition III.1. *Let the equations of kind $\alpha_s(x)=b$, $b\in\mathbb{K}$ have exactly one solution. Then map F_s is invertible.*

Proof: It is easy to check that if $F_s(\bar{x})=\bar{y}$ then $F_s^{-1}(\bar{y})=\bar{x}$. It is easy to see that $f_1(x_1,x_2,\dots,x_n)=\alpha_s(x_1)$. Let p be some point from P_n and $F_n(\mathrm{p})=(c_1,c_2,\dots,c_n)$ (point or line). Then the equation $\alpha_s(x_1)=c_1$ has a unique solution η. So we can compute $\eta_1=\alpha_1(\eta)$, $\eta_2=\alpha_2(\eta)$, …, $\eta_{s-1}=\alpha_{s-1}(\eta)$.

We can compute the chain $\mathrm{c}=(c_1,c_2,\dots,c_n)$, $N_{\eta_{s-1}}(\mathrm{c})=\mathrm{c}_1$, $N_{\eta_{s-2}}(\mathrm{c}_1)=\mathrm{c}_2$, …, $N_{\eta_1}(\mathrm{c}_{s-2})=\mathrm{c}_{s-1}$, $N_\eta((\mathrm{c}_{s-1}))=\mathrm{c}_s=(p_1,p_2,\dots,p_n)$ with $\eta=p_1$. So F_n is a bijection. ■

Notice that $N^s_{\mathrm{a}(x)}$ for $\mathrm{a}(x)$ of kind $\alpha_1(x)=\beta_1(x)$, $\alpha_2(x)=\beta_2(\alpha_1(x))$, $\alpha_3=\beta_3(\alpha_2(x))$, …, $\alpha_s(x)=\beta_s(\alpha_{s-1}(x))$ is a composition of ${N^1}_{\beta_1(x)}$, ${N^1}_{\beta_2(x)}$, …, ${N^1}_{\beta_s(x)}$. In this case invertibility of each $\beta_i(x)$, $i=1,2,\dots,s$ guarantees the bijectivity of $N^s_{\mathrm{a}(x)}$. We refer to such case as recurrently defined string.

Let L_1 and L_2 be sparse affine bijective transformation of the affine space (free module in other terminology) $\mathbb{K}^n$

$$\begin{aligned}L_1=T_{A,b}:\bar{x}&\longrightarrow\bar{x}A+b,\\ L_2=T_{C,d}:\bar{x}&\longrightarrow\bar{x}C+d,\end{aligned}$$

where $A=\left[a_{i,j}\right]$ and $C=\left[c_{i,j}\right]$ are $n\times n$ matrices with $a_{i,j},c_{i,j}\in\mathbb{K}$. It is clear that

$$\begin{aligned}L_1^{-1}=T^{-1}_{A,b}&=T_{A^{-1},-bA^{-1}},\\ L_2^{-1}=T^{-1}_{C,d}&=T_{C^{-1},-dC^{-1}}.\end{aligned}$$

Let F_n be a polynomial map of $\mathbb{K}^n$ to itself. We refer to $G_n=\tau_L F_n\tau_R$ as affine deformation of F_n.

Symmetric algorithm

We can use the data on the graph $T(n,\mathbb{K})$, the symbolic computation given by the string $\mathrm{a}=\mathrm{a}(x)$ of polynomials $\alpha_1(x)$, $\alpha_2(x)$, …, $\alpha_s(x)$, where $\alpha_s(x)$ is a bijective map of $\mathbb{K}$ to itself and affine transformations L_1 and L_2 in the following encryption scheme.

Correspondents Alice and Bob agree on a private encryption key

$$K_p=(L_1,L_2,\alpha=(\alpha_1,\alpha_2,\dots,\alpha_s)),$$

and keep the key in secret. Messages are written using characters belonging to the alphabet $\mathbb{K}$. So the plainspace is K^n and its elements must be treated as points (or lines) of the graph. To encrypt they use the composition

$$L_1\circ N^s_{\mathrm{a}}\circ L_2.$$

Notice that the computation has to be executed in numerical level:

1) Correspondent Alice writes plaintext $\mathrm{p}=(p_1,p_2,\dots,p_n)$ and treats it as point of the bivariate graph.
2) She computes parameters $\mu_i=\alpha_i(v_1)$ for $i=1,2,\dots,s$.
3) She computes p_0 as $L_1(\mathrm{p})$, p_1 as $N_{\mu_1}(\mathrm{p}_0)$, p_2 as $N_{\mu_2}(\mathrm{p}_1)$, …, p_s as $N_{\mu_s}(\mathrm{p}_{s-1})$.
4) She computes the ciphertext c as $L_2(\mathrm{p}_s)$.proo

Alice and Bob can use their knowledge about triple (L_1, L_2, a) for the decryption. Let us assume that Bob receives the ciphertext c from Alice. To decrypt the ciphertext Bob proceeds as follows:

1) He has to compute c_0 as ${L_2}^{-1}(\mathrm{c})$.
2) He treats the string of coordinates of this tuple as a vertex of the graph, which is a point in case of even s or the line in case of odd s with coordinates c_1^0, c_2^0, …, c_n^0.
3) Bob must find a solution η of $\alpha_s(x)=c_1^0$ and form a string $\eta_0=\eta$, $\eta_1=\alpha_1(\eta)$, $\eta_2=\alpha_2(\eta)$, …, $\eta_{s-1}=\alpha_{s-1}(\eta)$.
4) He computes c_1 as $N_{\eta_{s-1}}(\mathrm{c}_0)$, c_2 as $N_{\eta_{s-2}}(\mathrm{c}_1)$, …, c_s as $N_{\eta_0}(\mathrm{c}_{s-1})$.
5) He computes the plaintext p as ${L_1}^{-1}(\mathrm{c}_s)$.

Remark III.2. *In the case of identity maps L_1 and L_2 one can try Dijkstra's algorithm for finding the shortest path between plaintext and ciphertext. Notice that its complexity is $O(v \log v)$, but here v is exponential q^n. Therefore we get worse complexity even than brute force search via the key space.*

In the case of recurrently defined symbolic computation as above the encryption bijective map is $F_s = L_1 {N^1}_{\beta_1(x)} {N^1}_{\beta_1(x)} \dots {N^1}_{\beta_s(x)} L_2$. As we already see, this encryption transformation is equivalent to $L_1 {N^s}_{\mathrm{a}(x)} L_2$, where $\mathrm{a}(x) = (\beta_1(x), \beta_2(\beta_1(x)), \dots, \beta_s(\beta_{s-1}(\dots(\beta_1(x)))))$. Recurrently defined symbolic computation is an example of the polynomial map with an invertible decomposition in the sense of [31]. It has various applications in the development of multivariate key exchange protocols and asymmetric multivariate algorithm. The most popular case of implementation is related to graphs $D(n, \mathbb{K})$ (see [1, 21]) and $A(n, \mathbb{K})$ (see [22, 23]), where $\mathbb{K}$ is a finite field of arithmetical rings $\mathbb{Z}_m$ and strings of kind $\beta_1 = x + d_1$, $\beta_2 = x + d_2$, $\dots$, $\beta_s = x + d_s$, where $d_i + d_{i+1}$, $i = 1, 2, \dots, s-2$ are regular elements of the ring $\mathbb{K}$. We refer to such case as shifting encryption.

Let us consider the case of strong symmetric encryption, when the function is $\alpha_s(x) = ax+b$, with a regular (invertible) element of $\mathbb{K}$. In this case it is easy to show that degrees of encryption map F_n and decryption map ${F_n}^{-1}$ are the same. The advantage of this case is its universality. One can implement it in case of arbitrary chosen finite ring $\mathbb{K}$.

IV. On the properties of bivariate graph based bijective encryption maps

The girth G of simple graph G is the length of its shortest cycle. As it was established in [27] the girth of the graph $D(n, \mathbb{F}_q)$ is $\geq n+5$. So in the case of shifting encryption the map with the password $x + d_1$, $x + d_2$, $\dots$, $x + d_s$, $s < n+5$ the encryption map F_n has no fixed points. So ciphertext is always different from the plaintext. Let us consider deformated shifting encryption of kind $\tau_L F_n \tau_R$. We assume that affine maps τ_L and τ_R are fixed. Correspondents are able to change string $d_1, d_2, \dots, d_s$ for another one.

We assume that $d_i + d_{i+1} \neq 0$ for $i = 1, 2, \dots, s-2$. Such choice means that encryption map corresponds to the path of length s. The inequality $g(D(n, q)) \geq n + 5$ implies that different strings of length $s < (n+5)/2$ produce different ciphertexts. So even in the case when τ_L and τ_R are known to adversary the complexity of attacks without an access to unencrypted information is bounded from below by $q^{(n+5)/2}$.

In [44] these results were generalized for the case of general commutative ring $\mathbb{K}$. Let $\mathbb{M}$ be a multiplicative subset of $\mathbb{K}$, i. e. $\mathbb{M}$ is closed under the ring multiplication and it does not contain 0. We say that a string $d_1, d_2, \dots, d_s$ is $|\mathbb{M}|$-regular if $d_i + d_{i+1} \in \mathbb{M}$ for $i = 1, 2, \dots, s-2$. It was proven that different M-regular strings of length $s < (n+5)/2$ produce distinct ciphertexts from the same plaintext. So in the case of $|\mathbb{K}| = k$, $|\mathbb{M}| = m$ the resistance to attacks without access to unencrypted data is bounded from below by $mk^{(n+5)/2-1}$.

It was proven that graphs $A(n, \mathbb{F}_q)$ form a family of graphs of increasing girth $h(n)$ that tends to infinity as n grows. The speed of growth of $h(n)$ needs further evaluation. In [44] it was proven that different $|\mathbb{M}|$-regular strings of length $s < n$ produce different encryption maps.

Results on $|\mathbb{M}|$-regular strings of length restricted maps are obtained in terms of dynamical systems corresponding to graphs $D(n, \mathbb{K})$ and $A(n, \mathbb{K})$.

Let us assume that maps τ_L and τ_R are identities and consider the groups of transformations $GD(n, \mathbb{K})$ and $GA(n, \mathbb{K})$ generated by shifting encryption maps corresponding to strings of even length. In [40] was proven that all elements of $GD(n, \mathbb{K})$ are cubical transformations of affine spaces P_n and L_n. Similar result for $GA(n, \mathbb{K})$ is stated in [44]. As it follows instantly from this result transformation $F'_n = \tau_L F_n \tau_R$ and its inverse are cubical transformations.

The cryptanalytic corollary of this statement is justification of linearization attacks on stream ciphers corresponding to stream ciphers based on graphs $D(n, \mathbb{K})$ and $A(n, \mathbb{K})$.

Let correspondents use the transformation F'_n. The adversary has knowledge on the general scheme of open algorithm but not on the data for τ_L and τ_R and shifting string. So he knows about cubic nature of encryption. We assume that he has access to the unencrypted information and is able to intercept quite many pairs of kind (p, c), where p is plaintext and c corresponding ciphertext.

Then adversary writes G_n which is a formal cubical map in standard form with the unknown coefficients in front of monomial terms. He or she is able to solve system of $O(n^3)$ equations of kind $G_n(\mathrm{c}) = \mathrm{p}$ and restore the map G_n. So adversary could control the communication channel. The complexity of such direct linearization attack is $O(n^{10})$.

V. On the implementation of graph based stream cipher based on non bijective maps

Let us describe an implemented algorithm, which can run in the case of arbitrary commutative ring $\mathbb{K}$ and arbitrary bivariate graph $T(n, \mathbb{K})$. We slightly modify the above described symmetric algorithm based on bivariate graphs $T(n, \mathbb{K})$ which is not a case of shifting encryption. Firstly, we take a symbolic computation for string $\mathrm{a} = \mathrm{a}(x) = (\alpha_1(x), \alpha_2(x), \dots, \alpha_s(x))$, with $\alpha_i(x) = x^d + d_s$, $i = 1, 2, \dots, s$ where d is mutually prime with the order of $\mathbb{K}^*$. So equation $x^d + d_s = c$, $x \in \mathbb{K}^*$ has at most one solution. We take L_1 as an affine bijective transformation of kind $x_1 \to x_1 + x_2 + \dots + x_n$, $x_2 \to l_2(x_1, x_2, \dots, x_n)$, $x_3 \to l_3(x_1, x_2, \dots, x_n)$, $\dots$, $x_n \to l_n(x_1, x_2, \dots, x_n)$, where l_i are linear functions from $K[x_1, x_2, \dots, x_n]$. Correspondents will use the plainspace

$$\Omega = \{(x_1, x_2, \dots, x_n) | x_1 + x_2 + \dots + x_n \in \mathbb{K}^*, \\ x_i \in \mathbb{K}, i = 1, 2, \dots, n\}.$$

They will use $L_1 N^s_{\mathrm{a}(x)} L_2$ as encryption map. To execute computation in time $O(n)$ they take finite parameter s and use loaded tables for $\alpha_i(x)$, $i = 1, 2, \dots, s$ (one dimensional arrays $a_i(x)$, $x \in \mathbb{K}^*$). So they will compute $L_1(\mathrm{p}) = \mathrm{v} = (v_1, v_2, \dots, v_n)$, form sequence $\mu_i = \alpha_i(v_1)$, $i = 1, 2, \dots, s$

TABLE I
ENCODING AND DECODING TIME

		$A(n,\mathbb{K})$		$D(n,\mathbb{K})$		$\widetilde{D(n,\mathbb{K})}$	
Password	Filesize	Enc	Dec	Enc	Dec	Enc	Dec
3	1K	0.0021	0.0029	0.0030	0.0026	0.0039	0.0041
	10K	0.0217	0.0253	0.0234	0.0249	0.0322	0.0366
	50K	0.1030	0.1338	0.1034	0.1423	0.1572	0.1859
	100K	0.2158	0.2701	0.2115	0.2683	0.3309	0.3800
	500K	1.2202	1.3863	1.0432	1.3556	1.6161	1.9323
	1M	2.1955	2.8346	2.1452	2.7285	3.2809	3.9029
	10M	21.9597	27.4227	21.3803	26.6821	32.8819	38.3860
4	1K	0.0416	0.0033	0.0400	0.0032	0.0401	0.0047
	10K	0.0311	0.0320	0.0302	0.0360	0.0420	0.0466
	50K	0.1393	0.1639	0.1374	0.1580	0.2125	0.2366
	100K	0.2800	0.3314	0.2738	0.3280	0.4259	0.4816
	500K	1.4381	1.7109	1.3918	1.6541	2.1278	2.4159
	1M	2.9271	3.5035	2.8457	3.4055	4.3633	4.9664
	10M	29.5728	34.6022	28.6899	33.7773	43.7334	49.4341
5	1K	0.0402	0.0045	0.0336	0.0039	0.0437	0.0058
	10K	0.0355	0.0395	0.0382	0.0440	0.0533	0.0596
	50K	0.1764	0.2038	0.1718	0.1909	0.2589	0.2876
	100K	0.3510	0.4097	0.3391	0.3922	0.5243	0.5781
	500K	1.7778	2.0589	1.7237	2.0015	2.7088	3.0049
	1M	3.6421	4.2418	3.5507	4.1302	5.4671	6.0630
	10M	37.3170	42.0697	36.2427	40.9556	55.1103	60.4248
6	1K	0.0445	0.0053	0.0412	0.0046	0.0445	0.0069
	10K	0.0426	0.0481	0.0453	0.0448	0.0705	0.0667
	50K	0.2132	0.2371	0.1987	0.2325	0.3123	0.3462
	100K	0.4176	0.4830	0.4069	0.4678	0.6303	0.6890
	500K	2.1494	2.4572	2.0897	2.3724	3.2690	3.5826
	1M	4.3851	4.9386	4.2630	4.8109	6.7762	7.2091
	10M	47.8490	50.3557	42.6451	47.7372	65.8464	71.6511
7	1K	0.0434	0.0055	0.0435	0.0059	0.0487	0.0091
	10K	0.0477	0.0540	0.0475	0.0533	0.0754	0.0848
	50K	0.2437	0.2699	0.2324	0.2671	0.3651	0.3979
	100K	0.4903	0.5457	0.4751	0.5275	0.7315	0.7938
	500K	2.5089	2.8124	2.5655	2.7524	3.7086	4.0025
	1M	5.0959	5.7679	5.1230	5.6692	7.5859	8.2276
	10M	51.0014	56.3961	49.8712	54.9345	76.4318	87.4684

Algorithm 1 Encoding with graph $A(n,\mathbb{Z}_{256})$

Input: password $p = (p_0, p_1, \ldots, p_{k-1})$, message $m = (m_0, m_1, \ldots, m_{n-1})$
Output: encrypted message
$x = m$
for $i = 0, 1, ..., k-1$ **do**
  if $i \mod 2 = 0$ **then**
    $y_0 = (m_0^3 + p_i) \mod 256$
    for $j = 1, 2, \ldots, n$ **do**
      if $j \mod 2 \equiv 1$ **then**
        $y_j = (x_j - x_{j-1} \cdot y_0) \mod 256$
      else
        $y_j = (x_j - x_0 \cdot y_{j-1}) \mod 256$
  else
    $x_0 = (m_0^3 + p_i) \mod 256$
    $x_1 = (y_1 - y_0 \cdot x_0) \mod 256$
    for $j = 1, 2, \ldots, n$ **do**
      if $j \mod 2 \equiv 0$ **then**
        $x_j = (y_j + x_0 \cdot y_{j-1}) \mod 256$
      else
        $x_j = (y_j + x_{j-1} \cdot y_0) \mod 256$
if $k \mod 2 \equiv 1$ **then**
  return y
else
  return x

and compute recurrently $\mathrm{v}_i = N_{\mu_i}(\mathrm{v}_{i-1})$, $i = 1, 2, \ldots, s$. They form the ciphertext c as $L_2(\mathrm{v}_s)$.

To decrypt they will take $\mathrm{c}_0 = (c_1^0, c_2^0, \ldots, c_n^0)$ as ${L_2}^{-1}(\mathrm{c})$ and find a solution η for the equation $x^d + d_s = c_1^0$. Loaded table of values for ${\alpha_s}^{-1}$ will allow to find η fast. Next they form a string $\eta_0 = \eta$, $\eta_1 = \alpha_1(\eta)$, $\eta_2 = \alpha_2(\eta)$, $\ldots$, $\eta_{s-1} = \alpha_{s-1}(\eta)$. So users take string $\mathrm{c}_1 = N_{\eta_{s-1}}(\mathrm{c}_0)$, $\mathrm{c}_2 = N_{\eta_{s-2}}(\mathrm{c}_1)$, $\ldots$, $\mathrm{c}_s = N_{\eta_0}(\mathrm{c}_{s-1})$. Finally they get plaintext as $L^{-1}(\mathrm{c}_s)$.

The case of this symmetric algorithm appears as a private key for a cryptosystem introduced in [39] with the plainspace $\mathbb{Z}_m^n$.

We selected string of polynomials as $\alpha_i = x^d + d_i$, $d_i \in \mathbb{K}$, $i = 1, 2, \ldots, s$ and special linear transformations L_1 and L_2, given by the lists of linear forms.

We can theoretically evaluate degrees of encryption d_{enc} and decryption d_{dec}. In cases of graphs $D(n,\mathbb{K})$ and $A(n,\mathbb{K})$, these parameters are bounded below by some constants depending from parameters α_i, $i = 1, 2, \ldots, s$. We can select string of parameters and get d_{dec} large enough to make cryptoanalysis a difficult task. In case $\widetilde{D(n,\mathbb{K})}$ the degrees are even larger, they have size $O(n)$. Notice that direct linearization attacks are formally impossible because the encryption map is not a bijective one.

The implementation of the algorithms in the present work was done using the Python programming language, in particular version 2.7. The code doesn't use any out-of-the-box libraries for facilitating operations with matrices. The tests for measuring the processing time have been executed on a machine with Intel Core2 Duo CPU 9600 1.60GHz x 2, RAM memory 4.8 GB, operating with Ubuntu 16.04 LTS. The complexity of the algorithms is of order $O(sn)$, where s is the length of the password. In particular, we implement this stream cipher for case of $\mathbb{K} = \mathbb{Z}_{256}$ and $\alpha_i(x) = x^3 + d_i$ ($d = 3$ and $d_{\mathrm{dec}} = 43$), $i = 1, 2, \ldots, s$ without using loaded tables for functions. A description of the "nonlinear part" of encryption process, i. e. computation of N_{a}^s is presented below. We recommend a password for which d_2 and $d_i - d_{i+2}$, $i = 1, 2, \ldots, s-2$ are regular elements of the ring.

VI. CONCLUSION

The paper presents a class of stream ciphers defined in terms of graphs given by equations over the finite commutative ring $\mathbb{K}$. The algorithm has multivariate nature: plaintext is a tuple from the free module $\mathbb{K}^n$, key string is also an element of $\mathbb{K}^m$, the encryption map is polynomial transformation of $\mathbb{K}^n$ into itself. Users have options to vary parameters n and m and ring $\mathbb{K}$. If the parameter m is bounded by a constant, then the speed of numerical recurrent of encryption is $O(n)$. The key can be given as a sequence of polynomials in a single variable x. We observe results on simplest case of key strings $x + d_1$, $x + d_2$, $\ldots$, $x + d_s$ obtained by theoretical studies and via computer simulation in case of finite fields or arithmetical rings of kind $\mathbb{Z}_{2^m}$. In case of graphs $D(n,\mathbb{K})$ and $A(n,\mathbb{K})$ simple conditions on d_i ensure that different keys produce distinct ciphertexts and allow to estimate the complexity of adversary attacks without access to plaintext. In the above

Algorithm 2 Encoding with graph $D(n, \mathbb{Z}_{256})$

Input: password $p = (p_0, p_1, \ldots, p_{k-1})$, message $m = (m_0, m_1, \ldots, m_{n-1})$
Output: encrypted message
$x = m$
for $i = 0, 1, ..., k-1$ **do**
  if $i \mod 2 = 0$ **then**
    $y_0 = (m_0^3 + p_i) \mod 256$
    $y_1 = (x_1 + x_0 \cdot y_0) \mod 256$
    if $n \geq 2$ **then**
      $y_2 = (x_2 + x_0 \cdot y_1) \mod 256$
      if $n \geq 3$ **then**
        for $j = 3, 4, \ldots, n$ **do**
          if $j \mod 4 \equiv 3$ or $j \mod 4 \equiv 0$ **then**
            $y_j = (x_j + x_{j-2} \cdot y_0) \mod 256$
          else
            $y_j = (x_j + x_0 \cdot y_{j-2}) \mod 256$
  else
    $x_0 = (m_0^3 + p_i) \mod 256$
    $x_1 = (y_1 - y_0 \cdot x_0) \mod 256$
    if $n \geq 2$ **then**
      $x_2 = (y_2 - y_1 \cdot x_0) \mod 256$
      if $n \geq 3$ **then**
        for $j = 3, 4, \ldots, n$ **do**
          if $j \mod 4 \equiv 3$ or $j \mod 4 \equiv 0$ **then**
            $x_j = (y_j - y_0 \cdot x_{j-2}) \mod 256$
          else
            $x_j = (y_j - y_{j-2} \cdot x_0) \mod 256$
if $k \mod 2 \equiv 1$ **then**
  return y
else
  return x

Algorithm 3 Encoding with graph $\widetilde{D(n, \mathbb{Z}_{256})}$

Input: password $p = (p_0, p_1, \ldots, p_{k-1})$, message $m = (m_0, m_1, \ldots, m_{n-1})$
Output: encrypted message
$x = m$
for $i = 0, 1, ..., k-1$ **do**
  if $i \mod 2 = 0$ **then**
    $y_0 = (m_0^3 + p_i) \mod 256$
    $y_1 = (x_1 + x_0 \cdot y_0) \mod 256$
    if $n \geq 2$ **then**
      for $j = 2, 3, \ldots, n$ **do**
        if $j \mod 3 \equiv 2$ **then**
          $y_j = (x_j + (x_0 \cdot y_{j-1}) \mod 256$
        else if $j \mod 3 \equiv 0$ **then**
          $y_j = (x_j + x_{j-2} \cdot y_0) \mod 256$
        else
          $y_j = (x_j + x_{j-2} \cdot y_0 - x_0 \cdot y_{j-1}) \mod 256$
  else
    $x_0 = (m_0^3 + p_i) \mod 256$
    $x_1 = (y_1 - y_0 \cdot x_0) \mod 256$
    if $n \geq 2$ **then**
      for $j = 2, 3, \ldots, n$ **do**
        if $j \mod 3 \equiv 2$ **then**
          $x_j = (y_j - y_{j-1} \cdot x_0) \mod 256$
        else if $j \mod 3 \equiv 0$ **then**
          $x_j = (y_j - y_0 \cdot x_{j-2}) \mod 256$
        else
          $x_j = (y_j - y_0 \cdot x_{j-2} + y_{j-1} \cdot x_0) \mod 256$
if $k \mod 2 \equiv 1$ **then**
  return y
else
  return x

mentioned case encryption and decryption maps are cubical and adversary after the interception of $O(n^3)$ pairs of kind plaintext-ciphertext can conduct a linearization attack in time $O(n^{10})$. In case of $\widetilde{D(n, \mathbb{K})}$ the degree of both maps grows linearly with the growth of parameter n, which makes the search for the inverse map via linearization attacks a difficult task. Additionally, authors started investigation of bijective and non-bijective encryption maps with keys of kind $x^d + d_1$, $x^d + d_2$, ..., $x^d + d_s$, where $d > 1$.

In the non-bijective case the plainspace is large subset of $\mathbb{K}^n$ and the adversary has to restore the multivariate encryption transformation E and search for polynomial map E' such that EE' fixes each plaintext. Known methods do not allow to solve this task in polynomial time. Special case with high degree E' is implemented. Loaded tables for x^d allow a fast encryption of text even in case of large parameter d.

REFERENCES

[1] V. Ustimenko, *Coordinatisation of Trees and their Quotients*, in the Voronoj's Impact on Modern Science, Kiev, Institute of Mathematics, 1998, vol. 2, 125-152.

[2] D. Sharma, V. Ustimenko, *Special Graphs in Cryptography*, The Poster Papers Collection,Third International Workshop on Practice and Theory in Public Key Cryptography (PKC 2000), Melbourne Exhibition Centre, Australia, January 2000, p. 16- 19.

[3] V. Ustimenko, *CRYPTIM: Graphs as Tools for Symmetric Encryption*, Lecture Notes in Computer Science, Springer, LNCS 2227, Proceedings of AAECC-14 Symposium on Applied Algebra, Algebraic Algorithms and Error Correction Codes, November 2001, p. 278 - 286.

[4] V. Ustimenko, *Graphs with special arcs and cryptography* , Acta Applicandae Mathematicae (Kluwer) 2002, 74,117-153

[5] Yu, Khmelevsky, V. Ustimenko, *Walks on graphs as symmetric and asymmetric tools for encryption*, 2002, South Pacific Journal of Natural Studies, 2002, vol. 20, 23-41. www.usp.ac.fj/spjns

[6] Yu.Khmelevsky, M. Govorov, P. Sharma, V.Ustimenko, S. Dhanjal, *Security Solutions for Spatial Data in Storage (Implementation Case within Oracle 9iAS)*, Proceedings of 8th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2004) Orlando, USA, in July 18-21, 2004, pp 318-323.

[7] M. Govorov, Yu Khmelevsky, A. Khorev, V. Ustimenko, *Security Control for Spatial Warehouses*, Proceedings ofthe 21th International Cartographic Conference (ICC), Durban, South Africa, 2003, 1784-1794.

[8] Yu Khmelevsky, V Ustimenko, Practical aspects of the Informational Systems reengineering, The South Pacific Journal of Natural Science, volume 21, 2003, p.75-21 (www.usp.ac.fj/spjns/volume21).

[9] A. Tousene, V. Ustimenko, *CRYPTALL - a System to Encrypt All types of Data*, Notices of Kiev - Mohyla Academy , v. 23, 2004, pp 12-15.

[10] A. Tousene, V. Ustimenko,*Graph based private key crypto-system*, International Journal on Computer Research, Nova Science Publisher, vol.13, issue 4, 2005, 12p.

[11] V. Ustimenko, *On the extremal graph theory for directed graphs and its cryptographical applications*, Advances in Coding Theory and Cryptography, Series on Coding Theory and Cryptology, vol. 3, World Scientific, 181-200 (2007).

[12] V. Futorny, V. Ustimenko, *On small world semiplanes with generalised Schubert cells*, Acta Applicandae Mathematicae, 98, N1 (2007) 47–61 (with V. Futorny).

[13] V. Ustimenko, *On the graph based cryptography and symbolic computations*, Serdica Journal of Computing, Proceedings of International Conference on Applications of Computer Algebra 2006, Varna, N1 (2007).

[14] J. Kotorowicz, V. A. Ustimenko, *On the implementation of cryptoalgorithms based on algebraic graphs over some commutative rings*, Condenced Matters Physics, Special Issue: Proceedings of the international conferences “Infinite particle systems, Complex systems theory and its application”, Kazimerz Dolny, Poland, 2006, 11 (no. 2(54)) (2008) 347–360.

[15] V. Ustimenko, *On the hidden discrete logarithm for some polynomial stream ciphers*, International Multiconference on Computer Science and Informational Technology, 20-22 October 2008, Wisla, Poland, CANA Proceedings. 13 pp.

[16] S. Kotorowicz, V. Ustimenko, *On the properties of Stream Ciphers Based on Extremal Directed graphs*, In ”Cryptography Research Perspectives”, Nova Publishers, Ronald E. Chen (the editor), 2008.

[17] A. Touzene, V. Ustimenko, *Private and Public Key Systems Using Graphs of High Girth*,In ”Cryptography Research Perspectives”, Nova Publishers, Ronald E. Chen (the editor), 2008, pp.205-216.

[18] M. Klisowski, V. Ustimenko, *On the public keys based on the extremal graphs and digraphs*, International Multiconference on Computer Science and Informational Technology, October 2010, Wisla, Poland, CANA Proceedings, 12 pp.

[19] Y. Khmelevsky, Gaetan Hains, E. Ozan, Chris Kluka, V. Ustimenko and D. Syrotovsky) International Cooperation in SW Engineering Research Projects, Proceedings of Western Canadien Conference on Computing Education, University of Northen British Columbia,Prince George BC, May 6-7, 2011, 14pp.

[20] A. Touzene, V. Ustimenko, Marwa AlRaisi, Imene Boudelioua*Performance of Algebraic Graphs Based Stream-Ciphers Using Large Finite Fields*, Annalles UMCS Informatica AI X1, 2 (2011), 81-93.

[21] M. Klisowski, V. Ustimenko, *On the implementation of cubic public rules based on algebraic graphs over the finite commutative ring and their symmetries*, MACIS 2011: Fourth International Conference on Mathematical Aspects of Computer and Information Sciences, Beijing, 2011, 13 pp.

[22] M. Klisowski, U. Romanczuk, V. Ustimenko,*On public keys based on a new family of algebraic graphs* , Annalles UMCS Informatica AI X1, 2 (2011), 127 -141.

[23] J. Kotorowicz, U. Romanczuk, V. Ustimenko, *Implementation of stream ciphers based on a new family of algebraic graphs*, Proceedings of Federated Conference on Computer Science and Information Systems (FedCSIS), 2011, 13 pp.

[24] M. Klisowski, V. Ustimenko, *On the Comparison of Cryptographical Properties of Two Different Families of Graphs with Large Cycle Indicator*, Mathematics in Computer Science, 2012, Volume 6, Number 2, Pages 181-198.

[25] M. Polak, V. Ustimenko, *On stream cipher based on a family of graphs D(n, q) of increasing girth*, Albanian J. Math. 8 (2014), no. 2, 37-44.

[26] F. Lazebnik and V. Ustimenko, *Some Algebraic Constructions of Dense Graphs of Large Girth and of Large Size*, DIMACS series in Discrete Math- ematics and Theoretical Computer Science, V. 10 (1993), 75-93.

[27] F. Lazebnik, V. Ustimenko, *Explicit construction of graphs with an arbitrary large girth and of large size*, Discrete Appl. Math. , 60, (1995), 275 - 284.

[28] R. Wenger, *Extremal graphs with no C4, C6 and C10s*, 1991, J. Comb. Theory, Ser. B, 52(1),113-116.

[29] Ding J., Gower J. E., Schmidt D. S., *Multivariate Public Key Cryptosystems*, Springer, Advances in Information Security, V. 25, 2006.

[30] Polak M., Romańczuk U., Ustimenko V. and Wróblewska A., *On the applications of Extremal Graph Theory to Coding Theory and Cryptography*, Erdős Centennial, Proceedings of Erdős Centennial (EP 100), Electronic Notes in Discrete Mathematics,V43, P. 329–342 2013.

[31] Ustimenko V. A., *Explicit constructions of extremal graphs and new multivariate cryptosystems* Studia Scientiarum Mathematicarum Hungarica, Special issue ”Proceedings of The Central European Conference, 2014, Budapest”.

[32] V. A. Ustimenko, *On the cryptographical properties of extreme algebraic graphs*, in Algebraic Aspects of Digital Communications, IOS Press (Lectures of Advanced NATO Institute, NATO Science for Peace and Security Series - D: Information and Communication Security, Volume 24, July 2009, 296 pp.

[33] Ustimenko V. A., *On the flag geometry of simple group of Lie type and Multivariate Cryptography*, Algebra and Discrete Mathematics. V. 19. No 1. 2015. P. 130-144.

[34] Louis Goubin, Jacques Patarin, Bo-Yin Yang,*Multivariate Cryptography. Encyclopedia of Cryptography and Security*, (2nd Ed.) 2011, 824-828.

[35] Patarin J., *The Oil i Vinegar digital signatures*, Dagstuhl Workshop on Cryptography. 1997.

[36] Kipnis A., Shamir A., *Cryptanalisis of the Oil and Vinegar Signature Scheme* Advances in Cryptology - Crypto 96, Lecture Notes in Computer Science, V. 1462, 1996, P. 257–266.

[37] Bulygin S., A. Petzoldt A., and Buchmann J, *Towards provable security of the unbalanced oil and vinegar signature scheme under direct attacks*, In Guang Gong and KishanChand Gupta, editors, ”Progress in Cryptology - INDOCRYPT”, Guang Gong and Kishan Chand Gupta, editors, Lecture notes in Computer Science, V. 6498, 2010. P. 17–32.

[38] Romańczuk-Polubiec U., Ustimenko V, *On two windows multivariate cryptosystem depending on random parameters* Algebra and Discrete Mathematics. 2015. V. 19. No. 1. P. 101–129.

[39] V. Ustimenko, *On algebraic graph theory and non-bijective maps in cryptography*, Algebra and Discrete Mathematics, Volume 20 (2015). Number 1, pp. 152-170.

[40] A.Wróblewska, *On some properties of graph based public keys*, Albanian Journal of Mathematics, Volume 2, Number 3, 2008, 229-234, NATO Advanced Studies Institute: ”New challenges in digital communications”.

[41] U. Romańczuk, V. Ustimenko, *On regular forests given in terms of algebraic geometry, new families of expanding graphs with large girth and new multivariate cryptographical algorithms*, Proceedings of International conference ”Applications of Computer Algebra”, Malaga, 2013, p. 135-139.

[42] M. Klisowski, *Zwiekszenie bezpieczeństwa kryptograficznych algorytmów wielu zmiennych bazujacych na algebraicznej teorii grafów*, PhD thesis, Czestochowa, 2014.

[43] V. Ustimenko,*Linguistic Dynamical Systems, Graphs of Large Girth and Cryptography*, Journal of Mathematical Sciences, Springer, vol.140, N3 (2007) pp. 412-434.

[44] V. Ustimenko, U. Romańczuk, *On Dynamical Systems of Large Girth or Cycle Indicator and their applications to Multivariate Cryptography*, in ”Artificial Intelligence, Evolutionary Computing and Metaheuristics ”, In the footsteps of Alan Turing Series: Studies in Computational Intelligence, Volume 427/January 2013, pp. 231-256.

[45] V. Ustimenko, U. Romańczuk, *On Extremal Graph Theory, Explicit Algebraic Constructions of Extremal Graphs and Corresponding Turing Encryption Machines*, in ”Artificial Intelligence, Evolutionary Computing and Metaheuristics ”, In the footsteps of Alan Turing Series: Studies in Computational Intelligence, Volume 427/January 2013, pp. 257-285.

[46] U. Romańczuk, V. Ustimenko, *On the key exchange with matrices of large order and graph based nonlinear maps*, Albanian Journal of Mathematics, Volume 4, Number 4, pp. 203–211 (2010).

[47] V. Ustimenko, *Division algebras and Tits geometries*, DNAUSSR 296, No. 5 (1987), 1061-1065 (Russian)

[48] V. Ustimenko, *A linear interpretation of the flag geometries of Chevalley groups*, Kiev University, Ukrainskii Matematicheskii Zhurnal 42, No. 3 (March, 1990), 383-387; English transl.

[49] M. Polak, *On the applications of algebraic graph theory to coding*, PhD thesis, Maria Curie-Skłodowska, 2016.