Application of Pseudo-Memory Finite Automata for Information Encryption

Gulmira Shakhmetova[a], Zhanat Saukhanova[a], Nur Izura Udzir[b], Altynbek Sharipbay[a] and Nurgazy Saukhanov[c]

[a] L.N.Gumilyov Eurasian National University, Nur-Sultan, 010008, Kazakhstan
[b] Universiti Putra Malaysia, Selangor, 43400, Malaysia
[c] K.Zhubanov Aktobe Regional University, Aktobe, 030000, Kazakhstan

Abstract

Nowadays the development of information technologies bring to cryptologists not only opportunities to solve the most difficult cryptography classical tasks, also they give capacity to hacked well-known cryptosystems. Therefore, applying other areas of mathematical for modifications of information security methods is relevant task of research in cryptography. The theory of automata was considered as an alternative model for creating high-speed cryptosystems. In this paper, we survey existing works and concepts of finite automata cryptosystems with open key, its background and general algorithm of encrypting and decrypting process. According to the research carried out, it can be noted that in existing cryptosystems, finite automata of various types are used: finite automata of a general form, structural automata, finite automata with input-output memory of a special type, finite automata with pseudo-memory of a special type. The authors of the article were interested in the pseudo-memory automata that were used in FAPKC4. For a better understanding of the application of this type of finite automata in cryptography, the authors demonstrated an example of their application for encryption and decryption of information.

Keywords:

Cryptography, finite automata, cryptosystem, weakly invertible automata, pseudo-memory automata.

1. Introduction

Information technology has become an integral part of modern society life. The amount and value of information transmitted via the Internet increases every year, but the medium for data transfer is becoming more and more open, hence giving rise to the problem of protecting the information sent over unprotected communication channels. Today, the most reliable methods of protecting information are crypto-graphic methods [1]. The classic task of cryptographic methods is to hide the content of transmitted and stored data from unauthorized access. This problem is solved by data encryption, i.e. applying some mathematical transformations on the data, using a secret key, which is known only to the legitimate user.

Currently, there are many widely known cryptographic methods that are successfully used in practice. Many of these techniques are very computationally efficient. However, the development of

IntelITSIS'2021: 2nd International Workshop on Intelligent Information Technologies and Systems of Information Security, March 24–26, 2021, Khmelnytskyi, Ukraine
EMAIL: shakhmetova.gb@gmaill.com (G. Shakhmetova); saukhanova@mail.ru (Z. Saukhanova); izura@upm.edu.my (N. I. Udzir); sharalt@mail.ru (A. Sharipbay); snurgazi@mail.ru (N. Saukhanov)
ORCID: 0000-0002-7230-9475 (G. Shakhmetova); 0000-0001-9812-7736 (Z. Saukhanova); 0000-0002-0543-3329 (N. I. Udzir); 0000-0001-5334-1253 (A. Sharipbay); 0000-0002-1264-2460 (N. Saukhanov)

quantum computers, which allow to solve most of the classically difficult tasks, as well as the continuous improvement of cryptanalysis, lead to the emergence of new algorithms for hacking classical cryptographic systems. For example, scientists from the USA, the Netherlands and Australia discovered a serious vulnerability in the cryptographic library implemented in GnuPG, which allowed them to crack the 1024-bit RSA encryption [2]. This trend is of interest to cryptologists in the use of alternative mathematical models for the development of new and more advanced information security systems. In this article, an alternative method for designing cryptosystems, i.e. the theory of automata is considered. On the basis of various types of automata, such as Mealy automata, cellular automata, L-systems and others, some cryptosystems were created.

Automata theory, being a fundamental area of computer science, is engaged in the study of the recognition mechanisms of languages. The concept of an automaton can serve as a model object in a wide variety of problems, which makes it possible to apply the theory of automata in various scientific and applied research. This led to the wide use of the theory of automata in physics and cybernetics, chemistry and biology, economics and statistics, in cryptography and other sciences.

2. Background

The initiator of the use of finite automata in cryptography is a Chinese professor Tao Renji, who since the beginning of the 80s, together with Professor Chen Shihua, has been studying the theory of the invertibility of finite automata. This theory formed the basis of a new streaming cryptosystem with a public key, so in 1985 the finite automaton public-key cryptosystem, named Finite Automation Public Key Cryptosystems (FAPKC) was presented to the scientific world [3]. The first version of FAPKC0 [4], published in Chinese, uses linear components and was more demonstrative cryptoalgorithm. Versions of FAPKC1 and FAPKC2 using linear and nonlinear finite automata were available in English [5].

Ten years later, in 1995, some weaknesses in the open-text attacks, which were presented in [6] by Feng Bao, and Yoshihide Igarashi from Japan, were discovered in public-key cryptosystems FAPKC0 and FAPKC1. In [7], the authors Dai et al. introduced another way to break the cryptosystem FAPKC0. After the proposed options for attacks on the FAPKC0, FAPKC1, FAPKC2 cryptosystems by the authors Tao et al. an advanced asymmetric cryptosystem called FAPKC3 was introduced [8].

However, this algorithm was also cracked by the Finnish cryptologist Meskanen [9], whom described two methods for hacking some instances of the FAPKC3 cryptosystem, as well as ways to prevent these hacks. Finally, Tao and Chen presented a new version of the FAPKC4 public-key cryptosystem algorithm [10], which is crypto-resistant and still retains the advantages of the previously presented FAPKC, such as fast encrypting speed, a relatively short public key. This algorithm can be easily implemented, since it includes only logical operations. FAPKC4 was practically used in some local area networks in China [3].

In 2010, Chopuryan and Margarov proves that FAPKC3 is vulnerable to the against the chosen plaintext attack and to the exhaustive search attack as well. Therefore, modification version of FAPKC system was proposed in [11].

In 2011, a master student of De Montfort University, Leicester, UK, Sarshad Abubaker, under the direction of Dr. Kui Wu, offered his version of using finite automata in a cryptosystem, which is based on a 128-bit key using a DES-based key generation algorithm, known as DAFA (DES - Augmented Finite Automaton cryptosystem) [12].

In 2012, a new cryptographic algorithms based on Mealy/Moore automata and recursive functions were proposed [13] by S. Sri Lakshmi as a PhD work at the University of Technology named after Jawaharlal Nehru, India under the leadership of Professor B. Krishna Gandhi.

In 2016, Ivone de Fátima da Cruz Amorim published her doctoral thesis on “Linear Finite Transformers (LFT)” [14], where all characteristics of linear finite transducers and their reversibility are studied, and various examples are given in order to illustrate the proposed methods and concepts. Later in 2017, under her supervision, a master's work of Joana Barão Vieira [15] was published, in which the features of the formalization of the injectivity testing procedure and the construction of inverse finite memory trans-formers (linear and quasilinear) were disclosed.

In Russia, they also deal with the use of finite automata in cryptography. Agibalov [16] gives examples of using finite automata as cryptographic algorithms and their components, describes cellular automaton generators of pseudorandom sequences, cellular automaton hash functions, finite automaton symmetric and asymmetric ciphers, demonstrates functional equivalence of flow and automaton cryptosystems.

In [17, 18], the hardware implementation of the FAPKC cryptosystem based on field-programmable gate array (FPGA) was described, and the results of the study of the influence of the cryptosystem parameters on the dependence of the number of resources used and the performance of the FPGA were shown.

In Kazakhstan, scientists from the Eurasian National University were engaged in this research, and they created the hardware implementation of a public-key automated-field cryptosystem [19].

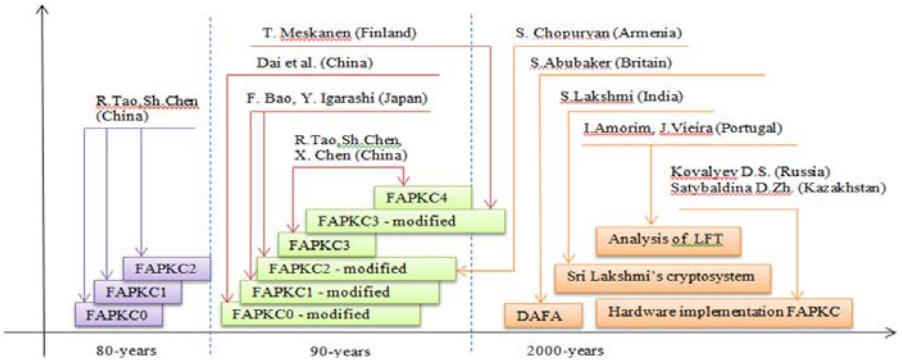

Figure 1: History of research development in the field of finite automata cryptography

According to Figure 1 it can be noted that with each decade, interest in the use of an alternative mathematical model in cryptosystems is increasing. This trend is due to the fact that in recent years the growth of information technologies has been significantly increased, which leads to the need to develop mechanisms for information protection.

3. Preliminaries

A finite automata (FA) is a mathematical abstract device that operates in discrete automata time. There are two types of automata: automata-recognizers and automata-transducers. We are only interested in automata-transducers, which convert the input sequence of words into an output sequence of the same length. In turn, the transforming automata can be divided into combinational finite automata (automata without memory) and sequential finite automata (automata with memory) [20]. This paper only considers the sequential finite automata.

Sequential FA – is a deterministic finite automata with a finite sequence of internal states that in any state reads an input symbol from the set X, outputs an output symbol from the set Y, and goes to another state, denoted by the symbol from the set S. If the symbols denoting the internal states of the automaton are stored in its internal memory, then this automaton is sometimes called a finite state machine with memory [21, 5p]. The formal definition of a sequential finite automaton is as follows [9]:

FA is a quintuple $M = < X, Y, S, \delta, \lambda >$, where: $X = \{x_1, x_2, \dots x_n\}$ – finite set of input symbols, $Y = \{y_1, y_2, \dots, y_m\}$ – finite set of output symbols, $S = \{s_1, s_2, \dots s_l\}$– finite set of internal states, $\delta: S \times X \rightarrow S$ – next state function or transition; $\lambda: S \times X \rightarrow Y$ – output function.

Let X^n be the set containing all finite words of length n in the alphabet X, X^ω be the set of words of infinite length in the alphabet X, and let ε be the empty word. Then the transition function $\delta: S \times X^n \rightarrow S$ and the output function $\lambda: S \times (X^n \cup X^\omega) \rightarrow Y$ can be expanded as:

$$\delta(s, \varepsilon) = s,\ \delta(s, \alpha x) = \delta(\delta(s, \alpha), x),$$
$$\lambda(s, \varepsilon) = \varepsilon,\ \lambda(s, x\alpha') = \lambda(s, x)\lambda(\delta(s, x), \alpha'),$$

where $s \in S, x \in X, \alpha \in X^n$ and $\alpha' \in X^n \cup X^\omega$.

In other words, FA M, being in the initial state $s(0)$ by reading the input sequence $x(0)x(1)\ldots$ passes a sequence of states $s(0)s(1)\ldots$ and produces an output sequence $y(0)y(1)\ldots$. The dependence between the input symbols, the states of the automaton M, and the output symbols in the discrete time i can be shown using the system of canonical equations:

$$\begin{cases} s(i+1) = \delta(s(i), x(i)) \\ y(i) = \lambda(s(i), x(i)) \end{cases} \quad i = 0,1,2,\ldots.$$

where $s(0)$ – initial state of the automaton, a $x(0) = \varepsilon$ and $y(0) = \varepsilon$.

Let there be given two finite automata $M = < X, Y, S, \delta, \lambda >$ and $M' = < X, Y, S', \delta', \lambda' >$. The FA $M = < X, Y, S, \delta, \lambda >$ is called weakly invertible with a delay τ, where τ is a nonnegative integer, if $\forall s \in S$ and $\forall x_i \in X, i = 0, 1, \ldots, \tau, x_0$ can be uniquely determined by the state s and the output function $\lambda(s, x_0 \ldots x_\tau)$.

For $\forall s \in S$ and $\forall s' \in S'$, if $\forall \alpha \in X^\omega, \exists \alpha_0 \in X^n$:

$$\lambda'(s', \lambda(s, \alpha)) = \alpha_0 \alpha \text{ and } |\alpha_0| = \tau,$$

then (s', s) is a pair with a delay τ ($\tau - pair$), or in other words, s' corresponds to s with a delay τ.

An automaton M' is said to be inverse with a delay τ to the automaton M if $\forall s \in S, \exists s' \in S'$ such that (s', s) is a $\tau - pair$ in $M' \times M$.

The finite-automaton model of a cryptosystem is based on the notion of a special form of weakly invertible finite automaton with a delay τ and composition of these automata. Will be given the following definitions, according to [22]:

If the function $\varphi: Y^k \times X^{h+1} \to Y$ for some integers $k, h \geq 0$, and if FA $M = < X, Y, Y^k \times X^{h+1}, \delta, \lambda >$ can be determined by

$$y(i) = \varphi(y(i-1), \ldots, y(i-k), x(i), \ldots, x(i-h)), \quad i = 0,1,\ldots,$$

i.e.

$$\delta(< y_{-1}, \ldots, y_{-k}, x_{-1}, \ldots, x_{-h} >, x_0) = < y_0, \ldots, y_{-k+1}, x_0, \ldots, x_{-h+1} >,$$
$$\lambda(< y_{-1}, \ldots, y_{-k}, x_{-1}, \ldots, x_{-h} >, x_0) = y_0,$$
$$y_0 = \varphi(y_{-1}, \ldots, y_{-k}, x_0, x_{-1}, \ldots, x_{-h}),$$

then M is called a (h, k) - order memory finite automaton and denoted by M_φ. Then h and k are called the input and output memory of the automaton M, respectively. In the case where $k = 0$, the automaton M_φ is called h-order input memory finite automaton.

Let function $f: Y^k \times U^{p+1} \times X^{h+1} \to Y$, and function $g: Y^k \times U^{p+1} \times X^{h+1} \to U$ for some integers $k, h \geq 0, p \geq -1$ and if finite automata $M_{f,g} = < X, Y, Y^k \times U^{p+1} \times X^h, \delta, \lambda >$ can be determined

$$y(i) = f\big(y(i-1), \ldots, y(i-k), u(i), \ldots, u(i-p), x(i), \ldots, x(i-h)\big),$$
$$u(i+1) = g\big(y(i-1), \ldots, y(i-k), u(i), \ldots, u(i-p), x(i), \ldots, x(i-h)\big), i = 0,1,\ldots$$

i.e.

$$\delta\big(< y_{-1}, \ldots, y_{-k}, u_0, \ldots, u_{-p}, x_{-1}, \ldots, x_{-h} >, x_0\big) = < y_0, \ldots, y_{-k+1}, u_1, \ldots, u_{-p+1}, x_0, \ldots, x_{-h+1} >,$$
$$\lambda\big(< y_{-1}, \ldots, y_{-k}, u_0, \ldots, u_{-p}, x_{-1}, \ldots, x_{-h} >, x_0\big) = y_0,$$
$$y_0 = f(y_{-1}, \ldots, y_{-k}, u_0, \ldots, u_{-p}, x_0, \ldots, x_{-h}),$$
$$u_1 = g(y_{-1}, \ldots, y_{-k}, u_0, \ldots, u_{-p}, x_0, \ldots, x_{-h}),$$

then M can be called (h, k, p) order pseudo-memory finite automata and denoted by $M_{f,g}$.

The automaton with memory in turn can be linear and nonlinear. If the functions defining the state machine are linear, then the state machine is linear. If any nonlinear function is added to the linear automaton, we obtain a nonlinear finite state machine with memory.

4. Public key cryptosystem based on finite automata

The basic concept of an asymmetric cryptosystem based on finite automaton is the use of a weakly invertible automaton with delay τ, which allows reproduction of an input sequence of characters by initial internal state and an output sequence of characters.

In the FAPKC cryptosystem, the public key is the composition of weakly invertible finite automata, whereas the private key contains their inverse automata. This cryptosystem can be used not only to encrypt and decrypt information, but also to sign and authenticate transmitted messages.

In the theory of numbers a large number can always be decomposed into simple factors, for which the order of their mutual arrangement in the product is not important. However in the theory of finite automata the order of arrangement of primitive automata in the composition is important. In other words, the composition of finite automata does not have the property of commutativity. Consequently, the problem of decomposition of compound finite automata into primitive components is as difficult as the factorization of the product of two large numbers. For that reason, this property allows creating ultra-reliable information security systems, which confirms the relevance and importance of creating cryptosystems based on finite automata [23].

As mentioned above, there are several versions of the asymmetric FAPKC cryptosystem. All versions of FAPKC have one common algorithm of the cryptosystem, the differences between them are in the generation of different types of finite automata that is used to encrypt/decrypt information, as presented in more detail in [24]. Next, we describe a general algorithm for building a cryptosystem, and take a closer look at the type of automata used in the version of FAPKC4.

In describing the general scheme of FAPKC we will rely on the work of [22], as follows:

Suppose that two users A and B want to exchange secret information, for this user A needs to generate a public key, which he will send to user B through an open channel for encrypting information, and a secret key, with which user A will decrypt the encrypted text received from user B. The public and private key are generated according to the following algorithm:

Two automata M_0 and M_1 are chosen randomly, for which it is easy (in polynomial time) to construct their inverse finite automata M_0^* and M_1^* with some delays τ_0 and τ_1, respectively. The composition of M_1 и M_0 - $C'(M_1, M_0)$ automata is constructed. Then $\tau = \tau_0 + \tau_1$ is determined. Select an arbitrary initial state of the automata $C'(M_1, M_0)$, which will be used in the beginning of encryption. The parts necessary for decryption are determined: $s_{1,d}^{out}$ and $s_{0,d}^{out}$. After the user's public key is composed, which consists of $\{C'(M_1, M_0), s_e, \tau\}$. The user's private key consists of $\{M_0^*, M_1^*, s_{1,d}^{out}, s_{0,d}^{out}, \tau_0, \tau_1\}$.

Encryption: User B adds arbitrary characters of length $x_{n+1} \cdots x_{n+\tau}$ to the end of the given plaintext $x_0 \cdots x_n$ and calculates the cipher text using the public key, $y_0 \cdots y_{n+\tau} = \lambda(s_e, x_0 \cdots x_{n+\tau})$. Then sends it to user A.

Decryption: User A receives the plain text in two steps. It first computes $x'_0 \cdots x'_{n+\tau-\tau_0}$ using M_0^* and $s_{0,d}^{out}$ and some part of s_e. Then finds $x_0 \cdots x_n$ using the automaton M_1^* and $s_{1,d}^{out}$.

The basic principles of encryption and decryption using a public key cryptosystem based on a state machine are shown in Fig 2 and Fig 3, respectively.

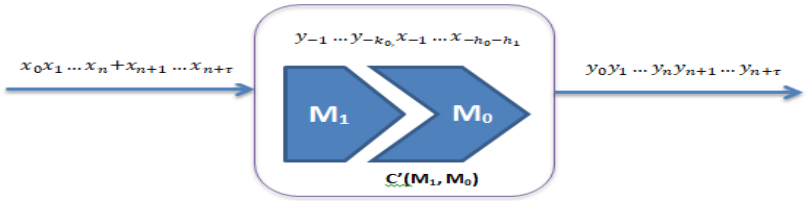

Figure 2: The principle of encryption in FAPKC

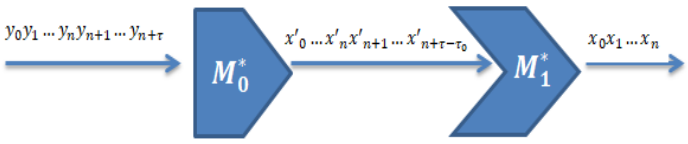

Figure 3: The principle of decryption in FAPKC

FAPKC have some advantages over widely used asymmetric cryptosystems, for instance, their encrypting speed faster that RSA's as their algorithm is based only on logical operations. In addition, the FAPKC cryptosystem can be attributed to stream encryption, which gives it an advantage in the encrypting and decrypting speed, because plaintext does not divided into blocks [4]. The disadvantages of the cryptosystem include the large size of the public key, as well as the problem of

generating random and equally probable keys, since the key space of the FAPKC algorithm is specified by the description of the properties of its elements [25].

According to [16] in order to keep the size of the public key within acceptable limits, it is necessary that the parameters of the cryptosystem are very small, as can be seen from the following Table 1 [26], where for some values of the parameters l, h_0, h_1 the corresponding sizes in bits N1 and N2 of the public key in FAPKC with $\tau_1 \le h_1, \tau_0 \le h_0$ and, respectively, with linear and nonlinear finite automata are demonstrated.

Table 1:
Size of keys FAPKC depends of parameters

l	7	7	5	5	3	3	3
(h_1, h_0)	(1,14)	(7,8)	(1,19)	(10,10)	(1,34)	(10,25)	(17,18)
N1	8281	32948	4075	20950	1593	8883	13041
N2	105840	414512	29850	181725	5400	34560	51192

5. Pseudo-memory automation for encryption/decryption information

In this section special types of automata is discussed, i.e. pseudo-memory automata that are used in the FAPKC4 cryptosystem. All definitions will be given according to Tao R. notation [22].

Let functions f_0 and g_0 single-valued mappings $Y^{k_0} \times U^{p_0+1} \times X^{h_0} \to Y$ and $Y^{k_0} \times U^{p_0+1} \times X^{h_0} \to U$, respectively, for some integers $k_0, h_0 \ge 0, p_0 \ge -1$ defining automata $M_0 =< X, Y, Y^{k_0} \times U^{p_0+1} \times X^{h_0}, \delta_0, \lambda_0 >$ with (h_0, k_0, p_0) pseudo-memory order:

$$y(i) = f_0\big(y(i-1), \dots, y(i-k_0), u(i), \dots, u(i-p_0), x(i), \dots, x(i-h_0)\big),$$
$$u(i+1) = g_0\big(y(i-1), \dots, y(i-k_0), u(i), \dots, u(i-p_0), x(i), \dots, x(i-h_0)\big),\text{i= 0,1, …}$$

i.e.

$$\delta_0\big(< y_{-1}, \dots, y_{-k_0}, u_0, \dots, u_{-p_0}, x_{-1}, \dots, x_{-h_0} >, x_0\big) = < y_0, \dots, y_{-k_0+1}, u_1, \dots, u_{-p_0+1}, x_0, \dots, x_{-h_0+1} >,$$
$$\lambda_0\big(< y_{-1}, \dots, y_{-k_0}, u_0, \dots, u_{-p_0}, x_{-1}, \dots, x_{-h_0} >, x_0\big) = y_0,$$
$$y_0 = f_0(y_{-1}, \dots, y_{-k_0}, u_0, \dots, u_{-p_0}, x_0, \dots, x_{-h_0}),$$
$$u_1 = g_0(y_{-1}, \dots, y_{-k_0}, u_0, \dots, u_{-p_0}, x_0, \dots, x_{-h_0}).$$

$M_0^* =< Y, X, X^{h_0} \times U^{p_0+1} \times Y^{\tau_0+k_0}, \delta_0^*, \lambda_0^* >$ - finite automata with pseudo-memory order $(\tau_0 + k_0, h_0, p_0)$ has the following form:

$$x(i) = f_0^*\big(x(i-1), \dots, x(i-h_0), u(i), \dots, u(i-p_0), y(i), \dots, y(i-\tau_0-k_0)\big),$$
$$u(i+1) = g_0\big(y(i-\tau_0-1), \dots, y(i-\tau_0-k_0), u(i), \dots, u(i-p_0), x(i), \dots, x(i-h_0)\big),\text{i=0,1, …}$$

i.e.

$$\delta_0^*\big(< x_{-1}, \dots, x_{-h_0}, u_0, \dots, u_{-p_0}, y_{-1}, \dots, y_{-\tau-k_0} >, y_0\big) = < x_0, \dots, x_{-h_0+1}, u_1, \dots, u_{-p_0+1}, y_0, \dots, y_{-\tau-k_0+1} >,$$
$$\lambda_0^*\big(< x_{-1}, \dots, x_{-h_0}, u_0, \dots, u_{-p_0}, y_{-1}, \dots, y_{-\tau-k_0} >, y_0\big) = x_0,$$
$$x_0 = f_0^*(x_{-1}, \dots, x_{-h_0}, u_0, \dots, u_{-p_0}, y_{-1}, \dots, y_{-\tau-k_0}),$$
$$u_1 = g_0(y_{-\tau_0-1}, \dots, y_{-\tau_0-k_0}, u_0, \dots, u_{-p_0}, x_0, \dots, x_{-h_0}).$$

Next, we give the definition of the automata with the input memory:

Let there be functions f_1 and g_1 single-valued mappings $W^{p_1+1} \times X^{h_1} \to Y$ and $W^{p_1+1} \times X^{h_1} \to U$, respectively, for some integers $h_1 \ge 0, p_1 \ge -1$ then the automaton $M_1 =< X, Y, W^{p_1+1} \times X^{h_1}, \delta_1, \lambda_1 >$ with a pseudo-memory of order *(h_1,0,p_1)* can be defined as:

$$y(i) = f_1\big(w(i), \dots, w(i-p_1), x(i), \dots, x(i-h_1)\big),$$
$$w(i+1) = g_1\big(w(i), \dots, w(i-p_1), x(i), \dots, x(i-h_1)\big),\text{i = 0,1, …}$$

i.e.

$$\delta_1(< w_0, \dots, w_{-p_1}, x_{-1}, \dots, x_{-h_1} >, x_0) =< w_1, \dots, w_{-p_1+1}, x_0, \dots, x_{-h_1+1} >,$$
$$\lambda_1(< w_0, \dots, w_{-p_1}, x_{-1}, \dots, x_{-h_1} >, x_0) = y_0,$$
$$y_0 = f_1(w_0, \dots, w_{-p_1}, x_0, \dots, x_{-h_1}),$$
$$w_1 = g_1(w_0, \dots, w_{-p_1}, x_0, \dots, x_{-h_1}),$$

The automaton inverse to it is $M_1^* =< Y, X, X^{h_1} \times W^{p_1+1} \times Y^{\tau_1}, \delta_1^*, \lambda_1^* >$ with a pseudo-memory of order (τ_1, h_1, p_1) has the following form:

$$x(i) = f_1^*(x(i-1), \dots, x(i-h_1), w(i), \dots, w(i-p_1), y(i), \dots, y(i-\tau_1)),$$
$$w(i+1) = g_1(w(i), \dots, w(i-p_1), x(i), \dots, x(i-h_1)),\text{i = 0,1, …}$$

i.e.

$$\delta_1^*(< x_{-1}, \dots, x_{-h_1} w_0, \dots, w_{-p_1}, y_{-1}, \dots, y_{-\tau} >, y_0) =$$
$$< x_0, \dots, x_{-h_1+1}, w_1, \dots, w_{-p_1+1}, y_0, \dots, y_{-\tau+1} >,$$
$$\lambda(< w_0, \dots, w_{-p_1}, x_{-1}, \dots, x_{-h_1} >, y_0) = x_0,$$
$$x_0 = f_1^*(x_0, \dots, x_{-h_1}, w_0, \dots, w_{-p_1}, y_0, \dots, y_{-\tau_1}),$$
$$w_1 = g_1(w_0, \dots, w_{-p_1}, x_0, \dots, x_{-h_1}),$$

Then the composition of two automata $f_1: W^{p_1+1} \times X^{h_1} \to Y$ and $f_0: Y^{k_0} \times U^{p_0+1} \times X^{h_0} \to Y$ can be represented as a finite automaton $C'(M_1, M_0)$ where the output of the automaton M_1 is the input of the automaton M_0:

$$f_1 \circ f_0: Y^{k_0} \times U^{p_0+1} \times W^{h_0+p_1+1} \times X^{h_0+h_1} \to Y$$

Substituting the values of the automaton M_1 into the automaton M_0, we obtain the following:

$$y(i) = f_0(y(i-1), \dots, y(i-k_0), u(i), \dots, u(i-p_0),$$
$$f_1(w(i), \dots, w(i-p_1), x(i), \dots, x(i-h_1)), \dots,$$
$$f_1(w(i-h_0), \dots, w(i-h_0-p_1), x(i-h_0), \dots, x(i-h_0-h_1))),$$
$$u(i+1) = g_0\Big(y(i-1), \dots, y(i-k_0), u(i), \dots, u(i-p_0),$$
$$f_1(w(i), \dots, w(i-p_1), x(i), \dots, x(i-h_1)), \dots, f_1(w(i-h_0), \dots, w(i$$
$$-h_0-p_1), x(i-h_0), \dots, x(i-h_0-h_1))\Big),$$
$$w(i+1) = w_1(w(i), \dots, w(i-p_1), x(i), \dots, x(i-h_1)),\text{i = 0,1, …}$$

i.e.

$$\delta(< y_{-1}, \dots, y_{-k_0}, u_0, \dots, u_{-p_0}, w_0, \dots, w_{-h_0-p_1}, x_{-1}, \dots, x_{-h_0-h_1} >, x_0) =$$
$$< y_0, \dots, y_{-k_0+1}, u_1, \dots, u_{-p_0+1}, w_1, \dots, w_{-h_0-p_1+1} x_0, \dots, x_{-h_0-h_1+1} >,$$
$$\lambda(< y_{-1}, \dots, y_{-k_0}, u_0, \dots, u_{-p_0}, w_0, \dots, w_{-h_0-p_1}, x_{-1}, \dots, x_{-h_0-h_1} >, x_0) = y_0,$$
$$y_0 = f_0(y_{-1}, \dots, y_{-k_0}, u_0, \dots, u_{-p_0}, w_0, \dots, w_{-h_0-p_1}, x_{-1}, \dots, x_{-h_0-h_1}),$$
$$u_1 = g_0\Big(y_{-1}, \dots, y_{-k_0}, u_0, \dots, u_{-p_0}, f_1(w_0, \dots, w_{-p_1}, x_0, \dots, x_{-h_1}),$$
$$\dots, f_1(w_{-h_0}, \dots, w_{-h_0-p_1}, x_{-h_0}, \dots, x_{-h_0-h_1})\Big),$$
$$w_1 = w_1(w_0, \dots, w_{-p_1}, x_0, \dots, x_{-h_1}),$$

Considering this type of automaton, we can see that the function that determines the pseudo-memory of the automaton is not subjected to modifications when constructing the inverse automaton. Next, consider the example of the automata with pseudo-memory in the encryption and decryption of information.

We will further illustrate an example on how to apply pseudo-memory automata on encryption and decryption process. Take two linear pseudo-memory automata over a finite field F (F=GF(2)), and take the parameters *m, l, n* = 3, in which $W = F^m, X = F^l, Y = F^n$ are the vector columns of dimension *m, l, n* =3, respectively

Let $M_1 =< X, Y, W^{p_1+1} \times X^{h_1}, \delta_1, \lambda_1 >$ be a liner automata with pseudo-memory of order *(4,0,0)* with delay $\tau_0 =1$ and $M_0 =< Y, Z, Z^{k_0} \times U^{p_0+1} \times Y^{h_0}, \delta_0, \lambda_0 >$ be a linear automata with pseudo-memory of order *(1,1,0)* with delay $\tau_1 =1$. M_0 and M_1 are defined as follows:

$$M_0: z(i) = \begin{pmatrix}1&1&0\\0&0&1\\0&1&0\end{pmatrix} z(i-1) + \begin{pmatrix}0&1&0\\1&0&0\\0&0&1\end{pmatrix} u(i-1) + \begin{pmatrix}1&0&0\\0&0&0\\0&0&0\end{pmatrix} y(i) + \begin{pmatrix}0&1&0\\0&1&0\\0&0&1\end{pmatrix} y(i-1)$$

$u(i-1) = g_0\big(z(i-1), z(i-2)\big) = z(i-1) \cdot z(i-2)$

$$M_1: y(i) = \begin{pmatrix}1&0&1\\0&1&1\\1&0&0\end{pmatrix} w(i-3) + \begin{pmatrix}1&1&0\\0&1&0\\1&0&0\end{pmatrix} x(i) + \begin{pmatrix}1&0&0\\0&1&0\\1&1&1\end{pmatrix} x(i-1) + \begin{pmatrix}1&0&0\\0&1&0\\1&0&1\end{pmatrix} x(i-2)$$

where B_3 and B_4 – the zero matrix, *i = 0,1,2,3*

$w(i-3) = g_1\big(x(i-3), x(i-4)\big) = x(i-3) \cdot x(i-4)$

For functions u and w, operation $(\cdot)$ means componentwise multiplication of vector elements, i.e. $[a_2, a_1, a_0]^T \& [b_2, b_1, b_0]^T = [a_2 \& b_2, a_1 \& b_1, a_0 \& b_0]^T$, where $a_i, b_i \in GF(2)$.

Construct the composition of these two automata $C'(M_1, M_0)$ with delay $\tau = \tau_0 + \tau_1 = 2$:

$$z(i) = \begin{pmatrix}1&1&0\\0&0&1\\0&1&0\end{pmatrix} z(i-1) + \begin{pmatrix}0&1&0\\1&0&0\\0&0&1\end{pmatrix} u(i-1) + \begin{pmatrix}1&1&0\\0&0&0\\0&0&0\end{pmatrix} x(i) + \begin{pmatrix}0&1&0\\1&1&0\\1&0&0\end{pmatrix} x(i-1) + \begin{pmatrix}0&0&1\\1&0&1\\1&1&1\end{pmatrix} x(i-2) + \begin{pmatrix}1&1&1\\1&1&1\\1&0&1\end{pmatrix} x(i-3) + \begin{pmatrix}1&0&1\\0&0&0\\0&0&0\end{pmatrix} w(i-3) + \begin{pmatrix}1&1&1\\1&1&1\\1&0&0\end{pmatrix} w(i-4)$$

Then, the public key is $C'(M_1, M_0)$ and some initial state s, for example, $s = < z_{-1}, z_{-2}, u_{-1}, w_{-3}, w_{-4}, x_{-1}, x_{-2}, x_{-3}, x_{-4}, x_{-5} > =$

$$< \begin{vmatrix}1\\0\\1\end{vmatrix}, \begin{vmatrix}0\\0\\1\end{vmatrix}, \begin{vmatrix}0\\0\\1\end{vmatrix}, \begin{vmatrix}1\\0\\0\end{vmatrix}, \begin{vmatrix}0\\1\\0\end{vmatrix}, \begin{vmatrix}0\\1\\0\end{vmatrix}, \begin{vmatrix}1\\0\\1\end{vmatrix}, \begin{vmatrix}1\\1\\0\end{vmatrix}, \begin{vmatrix}0\\1\\1\end{vmatrix} >$$

Where $u_{-1} = z_{-1} \cdot z_{-2}, w_{-3} = x_{-3} \cdot x_{-2}, w_{-4} = x_{-4} \cdot x_{-5}$

For example, to encrypt $\alpha = x_0 x_1 x_2 = \begin{vmatrix}1\\0\\1\end{vmatrix}\begin{vmatrix}1\\1\\0\end{vmatrix}\begin{vmatrix}1\\1\\1\end{vmatrix}$.

But, since the automata $C'(M_1, M_0)$ has delay 2, we need to append the plaintext with two arbitrarily chosen symbols, say $\gamma = x_3 x_4 = \begin{vmatrix}0\\0\\0\end{vmatrix}\begin{vmatrix}1\\0\\0\end{vmatrix}$, and calculate the ciphertext

$\lambda(s, \alpha\gamma) = \begin{vmatrix}0\\1\\1\end{vmatrix}\begin{vmatrix}0\\1\\1\end{vmatrix}\begin{vmatrix}1\\0\\0\end{vmatrix}\begin{vmatrix}0\\1\\1\end{vmatrix}\begin{vmatrix}0\\1\\1\end{vmatrix}$.

For the decryption process, it is necessary to compute the inverses of the automata M_0 and M_1, which will be the private key. Inverse automata M_0 and M_1 are constructed according to the rules of Ra/Rb transformations [22]. The inverse transducers are defined by:

$$M_0^*: y(i) = \begin{pmatrix}0&1&0\\0&0&0\\0&0&0\end{pmatrix} y(i-1) + \begin{pmatrix}0&0&0\\1&0&1\\0&0&1\end{pmatrix} u(i-1) + \begin{pmatrix}0&1&1\\0&0&0\\0&0&0\end{pmatrix} u(i-2) + \begin{pmatrix}0&0&0\\0&1&1\\0&0&1\end{pmatrix} z(i) + \begin{pmatrix}1&0&1\\0&1&1\\0&1&0\end{pmatrix} z(i-1) + \begin{pmatrix}1&0&0\\0&0&0\\0&0&0\end{pmatrix} z(i-2)$$

$u(i-1) = g_0\big(z(i-1), z(i-2)\big) = z(i-1) \cdot z(i-2)$

$u(i-2) = g_0\big(z(i-2), z(i-3)\big) = z(i-2) \cdot z(i-3)$

$$M_1^*: x(i) = \begin{pmatrix}1&1&0\\0&1&0\\0&1&1\end{pmatrix} x(i-1) + \begin{pmatrix}1&1&0\\0&1&0\\0&0&0\end{pmatrix} x(i-2) + \begin{pmatrix}0&0&0\\0&0&0\\0&1&0\end{pmatrix} w(i-2) + \begin{pmatrix}1&1&0\\0&1&1\\0&0&0\end{pmatrix} w(i-3) + \begin{pmatrix}0&0&0\\0&0&0\\1&1&1\end{pmatrix} y(i) + \begin{pmatrix}1&1&0\\0&1&0\\0&0&0\end{pmatrix} y(i-1)$$

$w(i-2)= w_1\big(x(i-2), x(i-3)\big) = x(i-2)\cdot x(i-3)$
$w(i-3)= w_1\big(x(i-3), x(i-4)\big) = x(i-3)\cdot x(i-4)$

In the first step of decryption we use M_0^* and $s_{0,d}^{out}$.

Calculate $y_0y_1y_2y_3= \lambda_0^*(< y_{-1}, u_{-1}, u_{-2}, z_0, z_{-1} >, z_1z_2z_3z_4)$.

Where $y_{-1} = \lambda_1(< w_{-1}, x_{-2}, x_{-3} >, x_{-1}) = \begin{vmatrix}1\\0\\0\end{vmatrix}$, $w_{-1} = x_{-4}\cdot x_{-5} = \begin{vmatrix}1\\1\\0\end{vmatrix}\cdot\begin{vmatrix}0\\1\\1\end{vmatrix} = \begin{vmatrix}0\\1\\0\end{vmatrix}$

$u_{-1} = z_0\cdot z_{-1} = \begin{vmatrix}1\\1\\1\end{vmatrix}\cdot\begin{vmatrix}1\\0\\1\end{vmatrix} = \begin{vmatrix}1\\0\\1\end{vmatrix}$, $u_{-2} = z_{-1}\cdot z_{-2} = \begin{vmatrix}1\\0\\1\end{vmatrix}\cdot\begin{vmatrix}1\\0\\0\end{vmatrix} = \begin{vmatrix}1\\0\\0\end{vmatrix}$

$y_0y_1y_2y_3 = \begin{vmatrix}1\\1\\1\end{vmatrix}\begin{vmatrix}1\\1\\0\end{vmatrix}\begin{vmatrix}0\\0\\1\end{vmatrix}\begin{vmatrix}0\\0\\0\end{vmatrix}$.

In the second step of decryption we use M_1^* and $s_{1,d}^{out}$.

Calculate $x_0x_1x_2 = \lambda_1^*(< x_{-1}, x_{-2}, x_{-3}, x_{-4}, w_{-2}, w_{-3}, y_0 >, y_1y_2y_3y_4)$.

After which we will get plaint text: $\begin{vmatrix}1\\0\\1\end{vmatrix}\begin{vmatrix}1\\1\\0\end{vmatrix}\begin{vmatrix}1\\1\\1\end{vmatrix}$.

The above example is given to illustrate the operation of the pseudo-memory machine. Obviously, in practice, FA is much more complicated. Automatic linear or non-linear automata are used depending on how it is built. In a nonlinear finite state machine, the degree of the polynomial constituting FA is greater than one. For more information on how to build such types of automata, refer to work [22].

6. Conclusion

This paper considered applying classical automaton-theoretical models to cryptography problems. The basic principle of constructing a finite-automaton cryptosystem with public keys and the history of the development of this direction are considered, an example of the use of pseudo-memory automata in information encryption is demonstrated. Summing up, it can be noted that the use of high-speed automata allows creating a strong asymmetric cryptosystem based on finite automata with high encrypting speed. However, using big parameters in cryptosystem will increase the size of public keys. Thereby finite-automaton cryptosystem with size of the public key within acceptable limits, could be used as part of a combination cryptosystem, which consists of several encryption algorithms and models. An approach to designing a cryptographic algorithm using finite automata with pseudo-memory is analyzed, which, in contrast to existing algorithms, makes it possible to increase the security of the designed asymmetric cryptosystem based on finite automata. In the future, it is planned to develop software modules for generating pairs of cryptographic keys for an asymmetric cryptosystem based on finite automata with pseudo-memory.

7. References

[1] A. Brito, S. Soares, S. Villela, Metaheuristics in the Project of Cellular Automata for Key Generation in Stream Cipher Algorithms, in: Proceeding of the IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 2018. doi: 10.1109/CEC.2018.8477658

[2] D. Bernstein, J. Breitner, D. Genkin, L. Bruinderink, N. Heninger, T. Lange, Ch. Vredendaal, Y. Yarom, Sliding right into disaster: Left-to-right sliding windows leak, in: Proceeding of the 19th International Conference on Cryptographic Hardware and Embedded Systems, CHES, Taipei, Taiwan, 2017. doi: 10.1007/978-3-319-66787-4 27

[3] G. Khaleel, Sh. Turaev, I. Al-Shaikhli, M. Mohd Tamrin, An overview of cryptosystems based on finite automata, in: Proceeding of the Jour of Adv.Rev. on Scientific Research 27 1 (2016) 1-7

[4] R. Tao, Sh. Chen, A finite automaton public key cryptosystem and digital signatures. Chinese Journal of Computers 8 6 (1985) 401-409 [In Chinese]
[5] R. Tao, Sh. Chen, Two varieties of finite automaton public key cryptosystem and digital signatures. Journal of computer science and technology 1 1 (1986) 9-18
[6] F. Bao, Y. Igarashi, Break finite automata public key cryptosystem. International Colloquiumon Automata, Languages, and Programming, Springer Berlin Heidelberg, pp. 147-158, 1995. doi: 10.1007/3-540-60084-1_70
[7] D. Dai, K. Wu, H. Zhang, Cryptanalysis on a finite automaton public key cryptosystem. Science in China 39 (1996) 27–36
[8] R. Tao, Sh. Chen, X. Chen, FAPKC3: a new finite automaton public key cryptosystem. Journal of Computer science and Technology 12 4 (1997) 289-305
[9] T. Meskanen, On finite automaton public key cryptosystems. TUCS Technical Report, Turku, 2001.
[10] R. Tao, Sh. Chen, The generalization of public key cryptosystem FAPKC4, Chinese science bulletin 44 9 (1999) 784-790
[11] G. Margarov, S. Chopuryan, Modification of Finite Automata Public Key Cryptosystem. Journal of Information Security Research 1 2 (2010) 39-54
[12] S. Abubaker, Wu. Kui, DAFA - A Lightweight DES Augmented Finite Automaton Cryptosystem. SecureComm, LNICST 106 (2013) 1–18
[13] S. Lakshmi, On finite state machines and recursive functions – applications to cryptosystems. PhD thesis. Jawaharlal Nehru Technological University, India, 2012.
[14] I. Amorim, Linear Finite Transducers Towards a Public Key Cryptographic System. PhD thesis. Porto University, Portugal, 2016.
[15] J. Vieira, Finite Transducers in Public Key Cryptography. Master thesis. Porto University, Portugal, 2017.
[16] G. Agibalov, State machines in cryptography. Applied Discrete Mathematics. Appendix Mathematical methods of cryptography 2 (2009) 45-73. [In Russian]
[17] D. Kovalev, Implementation on the FPGA cipher FAPKC, in: Proceeding of the 10th Siberian Scientific School-Seminar with International Participation Computer Security and Cryptography, SIBECRYPT'11, Tomsk, Russia, pp. 33-34, 2011[In Russian]
[18] D. Kovalev, Optimization of implementations of finite automaton encryption systems on the FPGA, in: Proceedings of Multi-core processes, parallel programming, FPGA, signal processing systems, Barnaul, 2014, pp. 38-45 [In Russian]
[19] D. Satybaldina, A. Sharipbayev, A. Adamova, Implementation of the Finite Automaton Public Key Cryptosystem on FPGA, in: Proceeding of the 8th International Workshop on Security in Information Systems, 2011, pp.167-173.
[20] A. Sharipbaj, Automata models in cryptography, Bulletin of Al-Farabi Kazakh National University 3/1 90 (2016) 96-104 [in Russian]
[21] A. Ozhiganov, Theory of automata: Textbook, NIU ITMO, Saint Petersburg, 2013.
[22] R. Tao, Finite Automata and Application to Cryptography. Tsinghua University Press, 2009.
[23] A. Sharipbay, Zh. Saukhanova, G. Shakhmetova, N. Saukhanov, Application of finite automata in cryptography, in: Proceeding of the International Conference on Engineering & MIS, ENU, Nur-Sultan, Kazakhstan, 2019. doi: 10.1007/978-3-540-78257-5
[24] A. Sharipbay, Zh. Saukhanova, G. Shakhmetova, M. Saukhanova, Ontology of finite-automaton cryptography. Ontology of designing l 31 (2019) 36-49. doi: 10.18287/2223-9537-2019-9-1-36-49.
[25] D. Satybaldina, A Sadykov, A. Adamova, Software and hardware implementation of a cryptosystem based on finite automata, Bulletin of ENU. L.N. Gumilyov 2 (2011)
[26] X. Chen, The Invertability Theory and Application of Quadratic Finite Automata. Laboratory for Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China, Doctoral Thesis, 1996