ISSN 2519-2310

COMPUTER SCIENCE AND CYBERSECURITY

ISSUE 3(11) 2018

V. N. Karazin Kharkiv National University Publishing

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ
MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені В.Н.КАРАЗІНА
ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ имени В.Н. КАРАЗИНА
V.N. KARAZIN KHARKIV NATIONAL UNIVERSITY

КОМП'ЮТЕРНІ НАУКИ ТА КІБЕРБЕЗПЕКА
КОМПЬЮТЕРНЫЕ НАУКИ И КИБЕРБЕЗОПАСНОСТЬ
COMPUTER SCIENCE AND CYBERSECURITY
(CS&CS)

Issue 3(11) 2018

Заснований 2015 року

Міжнародний електронний науково-теоретичний журнал
Международный электронный научно-теоретический журнал
International electronic scientific journal

The journal publishes research articles on theoretical, scientific and technical problems of effective facilities development for computer information communication systems and on information security problems based on advanced mathematical methods, information technologies and technical means.

Journal is published quarterly.
Approved for placement on the Internet by Academic Council of the Karazin Kharkiv National University (Dec. 17, 2018, protocol No.13)

Editorial office:
Karazin Kharkiv National University
Svobody Sq., 6, office 315a, Kharkiv, 61022, Ukraine (*North building of University, 3th floor*)
Phone: +38 (057) 705-10-83
E-mail: *cscsjournal@karazin.ua*
Web-page: *http://periodicals.karazin.ua/cscs (Open Journal System)*

Published articles have been internally and externally peer reviewed

TABLE OF CONTENTS

Issue 3(11) 2018

UDC 004.056.55

MATHEMATICAL MODEL AND METHODS OF PROCESSING BIOMETRIC IMAGES OF FINGERPRINTS

Emiliia Anishchenko, Serghii Rassomakhin

V. N. Karazin Kharkiv National University, 6 Svobody Sq., Kharkiv, 61022, Ukraine
emily9661@gmail.com, rassomakhin@karazin.ua

Reviewer: Vyacheslav Kharchenko, Doctor of Technical Sciences, Prof., Academician of the Academy of Sciences of Applied Radio Electronics, National Aerospace University named after. M. E. Zhukovsky, Kharkiv, Ukraine.
E-mail: v_s_kharchenko@ukr.net

Received on June 2018

Abstract: *Today, using of personal identification technologies based on biometric parameters for access to information resources is becoming topical in connection with the increase of informatization in modern society. Physical characteristics such as face, voice, retina or fingerprints are used to confirm personal identity. The most successful biometric identification technology is fingerprinting - a comparison of fingerprints. It is easy for using, reliable and there is no outside intrusion. The article will consider an analytical probabilistic model of formation and processing of fingerprints portraits taking into account possible natural factors that can distort the image of prints. Also, the model was processed to minimize the distortions' influence to approximate the distribution type of characteristic points to a similar distribution of the reference sample.*

Keywords: *biometry; model; methods; fingerprints.*

1 Introduction

Personal identification using of fingerprinting is a promising developing direction. Biometric information is unique, it can not be forgotten or lost. The presentation of information requires the physical personal presence. The method uses the uniqueness of the pattern of papillary patterns on the fingers of people.

The imprint obtained using the scanner converts into a digital code and then compares with previously entered sets of standards. Obtaining an electronic fingerprint with a clearly visible papillary pattern is a difficult task. The fingerprint is too small, it is necessary to use various methods to obtain its high-quality image. A lot of external factors can distort the patterns of prints during scanning the finger.

Today, all biometric technologies are probabilistic and none of them can guarantee the complete absence of errors.

The advantages of the fingerprint identification method are:

• The uniqueness of fingerprints. They are unique from each other as well as other fingerprints of any person. Even twins have fingerprints that are different.

• It is impossible to lose or forget fingerprints, as with passwords or PIN codes.

• Fingerprints do not change over time.

• Fingerprints have been used for many years to identify individuals. Therefore, it is possible to approve the developed algorithms using existing databases.

In each fingerprint you can define two types of signs - global and local. Global signs – fingerprint characteristics that you can see easily. Global signs include the image region, the core, the "delta" point, the line counter, the papillary pattern. Local signs called Minutiae are small unique points for each fingerprint, which are used to identify the individual. A fingerprint may have the same global signs but local signs are always unique. Therefore, the process of individual identification usually consists of two stages. The first step is the classification of fingerprints on global grounds using the databases for the division into classes and at the phase of authentication. The second stage is the fingerprint recognition (*identification*) on the comparison basis of the structure and coincidence coefficient of minutiae points.

2 Principle of fingerprint recognition

Depending on the obtained image of fingerprints quality we can identify some fingers surface characteristic features, which can later be used for identification purposes. If the image resolution obtained from the scanner is 300–500 dpi, a sufficiently large number of small details (*minutiaes*) can be distinguished in the fingerprint image. They can be divided into two automated types: endpoints - points where papillary lines finish; and branch points - points where the papillary lines split into two. In Fig. 1 is an example showing endpoints and branch points.

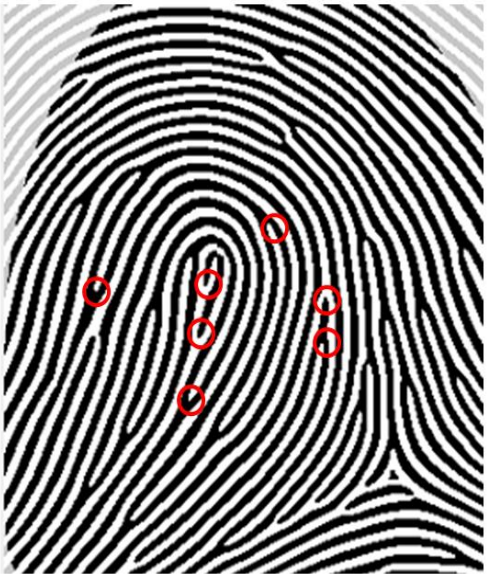


Fig. 1 - Endpoints and branch points

The fingerprint identification principle depends on the presence of the special points on the print - minutiaes. Each minutiae conditionally has own coordinates, direction and type (ending, branching, etc.). When algoritm has a set of singular points {(X, Y, θ, type), which were retrieved during registration it evaluates the point samples similarity and gives the result – "Identified" or "Not recognized". In the task of identification / verification of a person by fingerprint, the following three main stages can be distinguished which are modern algorithms characteristic:

1) processing of the original image;
2) isolation of minutiaes;
3) comparison of minutiaes' fingerprints.

As a rule, if it is not received electronically the fingerprint original image has bad quality (*lines are damaged, there are different distortions, etc.*).

The distortion types and distribution functions are not simple because of the causes multiple nature. All existing fingerprint recognition algorithms are closed for access, so there is no possibility to study the placement statistics and minutiaes distortion which would be sufficient for a clear problem solution. Therefore, the models presented below were obtained on the scanning processes heuristic analysis basis taking into account the possible errors nature.

3 Mathematical model of fingerprints characteristic points distribution

To develop an obtaining high-entropy data method based on various biometric images implementations the model has to describe the minutiae random distribution and arrival angles as well as the error realizations random distribution. According to the minutiae distribution portraits analysis the following *features* can be noted:

- point distribution density along the horizontal (X) and vertical (Y) axis has an approximately level character in the frame central part and slightly decreases to its edges;
- fingerprint center linear displacements on the horizontal and vertical do not mean the appearance of free from minuciae zones at the edges of the frame (*new points can get into the scanning field*);
- "arrival angles" distribution at characteristic points is approximately level in the range $[0,\ 2\pi]$.

To build a minuciae distribution model we will assume that:

- fingerprint portrait coordinates X, Y as well as the arrival angles values are normalized in the range [-0.5; +0.5], while the image geometric center has coordinates [0;0]. The portrait is placed in a unit square area covering all image plane;
- for the initial random numbers generation required to obtain the minutiaes coordinates on fingerprints implementations we use the sensor with uniformly distributed (*continuously*) numbers in the range [0;1]: $f(x_i, y_i) \sim \text{unif}[0,1],\ i \in 1 \ldots N$ where N is the number of minutiaes in the portrait –

random value in the range with [15; 60] with mathematical expectation $m_N = 25 \div 35$ and unimodal distribution.

Therefore, we can use the dependence shown in Fig. 2 for the formal *probability distribution density* (PDD) description f(x) and f(y).

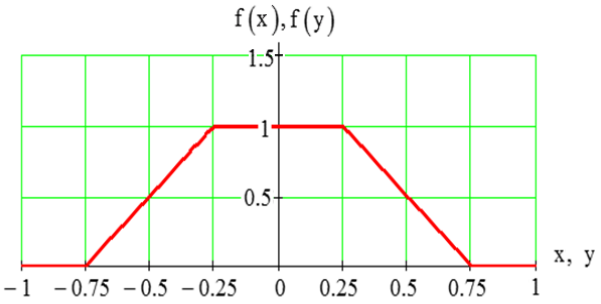


Fig. 2 – The minutiae coordinates probability density functions

To obtain minutiaes placement portraits test samples we need a random numbers source, distributed in accordance with the PDD (Fig. 2). Due to the distributions identity along the plane coordinates using the normalized unit portrait square in the future we will use only the function:

$$f(x) = \begin{cases} 2x + 1.5 & \text{if } -0.75 \le x < -0.25; \\ 1 & \text{if } -0.25 \le x \le 0.25; \\ -2x + 1.5 & \text{if } 0.25 < x \le 0.75; \\ 0 & \text{if } |x| > 0.75. \end{cases} \quad (1)$$

Using the inverse functions method:
if $z \sim \text{unif}[0,1]$ then a random variable x obtained by a functional transformation z of the form

$$x = \begin{cases} \sqrt{z} - 0.75 & \text{if } 0 \le z < 0.25; \\ z - 0.5 & \text{if } 0.25 \le z \le 0.75; \\ -\sqrt{1-z} + 0.75 & \text{if } 0.75 < z \le 1; \end{cases} \quad (2)$$

will have PDD (1).

Fig. 3 shows the functional transformation (2) statistical test histogram with the experiments equal number to 30,000 and the interval division [-0.75; +0.75] into 100 equal subintervals. The dashed line in fig. 3 shows the envelope line (1).

The generating random numbers resulting algorithm will be used in the future to obtain the prints normalized square portraits characteristic points coordinates.

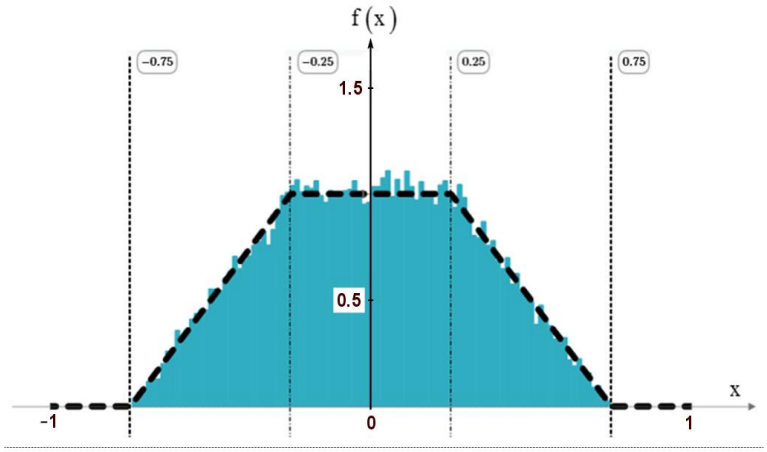


Fig. 3 – The sensor random coordinates statistical tests result

To generate a random number of characteristic points for the print portrait implementation we use discrete (integer) random variable model N: [15, 45] with a discrete normal truncated distribution and numerical characteristics:

- expectation $m_N \approx 30$;
- standard deviation $\sigma \approx 2 \div 5$.

To obtain a minutiae random number on the portrait implementation N, we use the unif[0,1] functional data transformation. We proceed to the uniformly distributed numbers discrete form using the integer rounding and centering operation:

$$z' = \text{round}(z) - 0.5, \text{ where } z \sim \text{unif}[0,1]. \quad (3)$$

Then, with limit to the number of terms equal to $m_N = 30$ the minutiae random number in the portrait can be defined as the sum

$$N = \sum_{i=1}^{30} z' + 30 \tag{4}$$

A discrete random variable can take integer values from a range [15, 45]. Than the truncated normal function of the PDD:

$$Q(N_i) = \binom{i}{30} \cdot \left(\frac{1}{2}\right)^{30}, \quad i \in [0,30], \; N_i \in [15,45] \tag{5}$$

Where $Q(N_i)$ is the probability that the minutes number in a portrait (taking into account the points masked outside the unit square) will be N_i.

The distribution type and numerical characteristics (5) are shown in fig. 4.

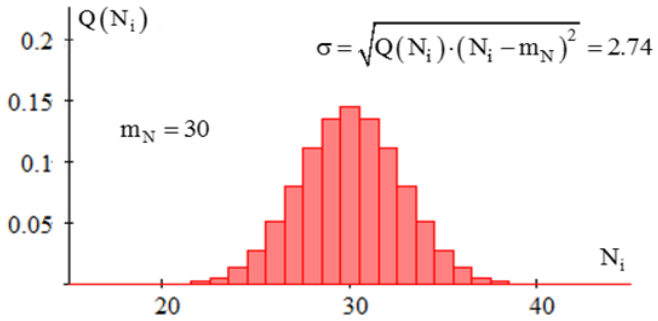


Fig. 4 – PDD normalized print portrait of the characteristic points number

To simulate the arrival angles random values normalized in a unit square at characteristic points, we use a random variable z ~ unif [0,1] uniformly distributed over a single segment:

$$\varphi = z. \tag{6}$$

The true arrival angle is determined based on the normalized value (6): $\Phi = 2\pi \cdot \varphi$.

Table 1 presents the normalized portrait simulation results based on the distributions (1), (4) and (6).

Table 1 - Portrait Matrix

№	X	Y	φ
1	0.21	-0.07	0.6
2	0.28	0.18	0.58
3	0.12	-0	0.49
4	0.19	0.31	0.74
5	0.07	-0.68	0.62
6	0.05	-0.12	0.8
7	-0.25	0.33	0.58
8	-0.01	0.42	0.91
9	0.17	0.15	0.73
10	0.12	0.23	0.67

4 Mathematical model of the characteristic point distortion

We can list the main factors that lead to differences in fingerprint image implementations:

- geometric center displacements, caused by a change with the object position in the scanning field;
- images rotation arising for the same reasons;
- "erasure" or the appearance of "false" points due to scanner algorithm incorrect settings or the foreign objects entry in the scanning field;
- points relative position drift due to errors in the recognition algorithm.

Geometric center displacement errors

To describe such errors distribution we use a PDD with a form (1) with a horizontally modified scale:

$$\Phi(d_y), \Phi(d_x) = \begin{cases} 12.5 \cdot d_x + 3.75 & \text{if } -0.3 \le d_x < -0.1; \\ 2.5 & \text{if } -0.1 \le d_x \le 0.1; \\ -12.5 d_x + 3.75 & \text{if } 0.1 < d_x \le 0.3; \\ 0 & \text{if } |d_x| > 0.3. \end{cases} \tag{7}$$

If z ~ unif [0,1], then for obtaining a random variable with distribution (7) we have to use the transformation:

$$d_x(z) = \begin{cases} \sqrt{0.16 \cdot z} - 0.3 & \text{if } 0 \le z < 0.25; \\ 0.4 \cdot z - 0.2 & \text{if } 0.25 \le z \le 0.75; \\ -\sqrt{0.16 \cdot (1-z)} + 0.3 & \text{if } 0.75 < z \le 1. \end{cases} \tag{8}$$

Errors due to "erasure" and adding "false" points

Assuming erasure errors independence distribution is described by the usual discrete binomial distribution:

$$P_E(k) = \sum_{i=0}^{k} \binom{i}{N} \cdot p_E{}^i \cdot (1-p_E)^{N-i} \quad (9)$$

Where $P_E(k)$- the probability of erasure no more than points in the portrait; N - the identified points number, determined by the distribution (5).

We generate a vector $Z = \{z_1, z_2, \cdots, z_N\}$. Based on the vector Z we calculate the erase vector the elements of which have a binary value and are obtained by a vector Z coordinates functional transformation:

$$e_i = \left\lfloor \frac{z_i}{1-p_E} \right\rfloor, \quad e_i = [0,1], \ i = 1 \ldots N \quad (10)$$

A probabilistic errors description associated with the false points appearance can be made on the Poisson distribution basis:

$$\Pr(K) = \frac{\lambda_A{}^K}{K!} e^{-\lambda_A}, \quad (11)$$

where Pr (K) is the false point appearance probability on the portrait implementation; λ_A – (Appearance) false point number empirically determined expectation in one portrait.

The approximation of random variable – the false points number in a portrait (11) is achieved using a sensor unif [0,1] and a usual binomial distribution as follows: an expectation is determined empirically by the false points number in one portrait λ_A: $0.1 \le \lambda_A \le 0.5$. A vector $Z = \{z_1, z_2, \cdots, z_M\}$ is generated with elements uniformly distributed in a unit interval z ~ unif [0,1]. Then, on the transformation basis similar to (10), we get an adding vector with binary elements obtained by the rule:

$$a_i = \left\lfloor \frac{z_i}{1-\frac{\lambda_A}{M}} \right\rfloor, \quad a_i = [0,1], \ i = 1 \ldots M. \quad (12)$$

The square length of the vector determines the false point number added to the portrait:

$$K = |A|^2 = \sum_{i=1}^{M} a_i. \quad (13)$$

The Poisson distribution approximation (11) will be more exact if the selected value M will be larger. For an acceptable approximation it suffices to require the inequality fulfillment of:

$$M \ge \lambda^{-1}. \quad (14)$$

Image rotation errors

The relationship between the arbitrary point coordinates $\begin{pmatrix} X \\ Y \end{pmatrix}$ in the original coordinate system XOY and the new point coordinates $\begin{pmatrix} X' \\ Y' \end{pmatrix}$ in the system X′OY′ deployed at an angle α is specified in matrix form with the following expression:

$$\begin{pmatrix} X' \\ Y' \end{pmatrix} = U(\alpha) \cdot \begin{pmatrix} X \\ Y \end{pmatrix}, \text{ where } U(\alpha) = \begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix}. \quad (15)$$

The portrait points' location rotation by an angle is achieved by converting the submatrix:

$$XY' = U(-\alpha) \cdot (XY)^T, \quad (16)$$

where $(XY)^T$ is the transposed submatrix XY.

The change in the vector associated with the rotation of the portrait is determined by the formula:

$$\varphi' = \left(\varphi - \frac{\alpha}{2\cdot\pi}\right) \bmod 1 - \left\lfloor \left(\varphi - \frac{\alpha}{2\cdot\pi}\right) \bmod 1 \right\rfloor, \quad (17)$$

where the operation extracts the fractional part from 'arg' taking into account a sign.

The matrix of the portrait turned to the angle α is determined with the submatrices and the vector union:

$$G' = \left((XY')^T \| \varphi'\right). \quad (18)$$

Image rotation errors are defined by the random rotation angle distribution. Empirical considerations allow to limit the range of possible values within the right angle

$$\alpha \in \left[-\frac{\pi}{4}, +\frac{\pi}{4}\right]. \quad (19)$$

The rotation random normalized angle implementation is obtained by a functional transformation of the form:

$$\alpha_H = \sum_{i=1}^{4}\left(\frac{z_i - 0.5}{16}\right). \quad (20)$$

Approximate normal PDD is:

$$f(\alpha_H) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\alpha_H^2}{2\sigma^2}\right), \quad \sigma = \sqrt{\frac{1}{3\cdot 256}}. \quad (21)$$

Fig. 5 shows the function form (21) (*dashed line*) and the approximation probability distribution histogram (20), obtained with the 10^5 tests. As we can see, using only four terms in the sum expression (20) provides a normal PDD approximation.

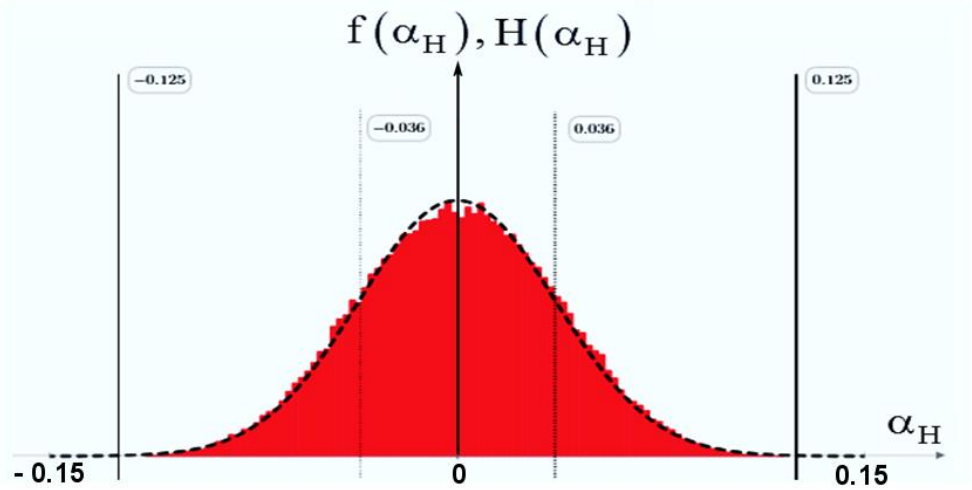


Fig. 5 – The PDD and the rotation normalized random angle PDD approximation histogram

5 Getting the original model portrait in the normalized unit square

The portrait implementation generation is performed with using the obtained functional transformations of a random variable uniformly distributed in a unit interval.

Determine the detected characteristic points. Using a software sensor with uniformly distributed numbers we get a vector of 15 elements: $Z=\{z_1, z_2, \ldots, z_{15}\}$, where $z_i \sim \text{unif}[0,1]$.

Table 2 – Z

№	z_i	№	z_i	№	z_i
1	0.001	6	0.174	11	0.989
2	0.193	7	0.71	12	0.119
3	0.585	8	0.304	13	0.009
4	0.35	9	0.091	14	0.532
5	0.823	10	0.147	15	0.602

Table 3 – Z′

№	z_i'	№	z_i'	№	z_i'
1	-0.5	6	-0.5	11	0.5
2	-0.5	7	0.5	12	-0.5
3	0.5	8	-0.5	13	-0.5
4	-0.5	9	-0.5	14	0.5
5	0.5	10	-0.5	15	0.5

Using the transformation (3) go to the vector containing the centered binary elements: $Z' = \text{round}(Z) - 0.5$.

Applying the transform (4) using elements z_i' gives a random number of minutes:

$$N = \sum_{1}^{15} z' + 15 = 16 .$$

For random coordinates generating of points on a plane we use the sensor unif[0,1] and get two vectors (Tables 4 and 5) which contains random numbers from the range [0,1]:

$$Z^1 = \left\{z_1^1, z_2^1, \cdots, z_N^1\right\}, \quad Z^2 = \left\{z_1^2, z_2^2, \cdots, z_N^2\right\}.$$

Table 4 – Z^1

№	Z_i^1	№	Z_i^1
1	0.376	9	0.458
2	0.677	10	0.744
3	0.009	11	0.599
4	0.276	12	0.735
5	0.588	13	0.572
6	0.838	14	0.152
7	0.485	15	0.425
8	0.744	16	0.517

Table 5 – Z^2

№	Z_i^2	№	Z_i^2
1	0.28	9	0.55
2	0.682	10	0.472
3	0.722	11	0.847
4	0.123	12	0.456
5	0.835	13	0.983
6	0.517	14	0.739
7	0.426	15	0.196
8	0.949	16	0.839

Applying the transformation (2) to the tables 4 and 5 elements we obtain two coordinate vectors:

Table 6 – X

№	x_i	№	x_i
1	-0.124	9	-0.042
2	0.177	10	0.244
3	-0.656	11	0.099
4	-0.224	12	0.235
5	0.088	13	0.072
6	0.347	14	-0.361
7	-0.015	15	-0.075
8	0.244	16	0.017

Table 7 – Y

№	y_i	№	y_i
1	-0.22	9	0.05
2	0.182	10	-0.028
3	0.222	11	0.359
4	-0.399	12	-0.044
5	0.343	13	0.619
6	0.017	14	0.239
7	-0.074	15	-0.307
8	0.525	16	0.349

Simulating the normalized arrival angles values in the minutiaes we use unif [0,1] and get the vector φ (Table 8):

Table 8 – φ

№	φ_i	№	φ_i	№	φ_i	№	φ_i
1	0.806	5	0.752	9	0.437	13	0.696
2	0.211	6	0.543	10	0.578	14	0.19
3	0.553	7	0.437	11	0.629	15	0.178
4	0.114	8	0.696	12	0.504	16	0.457

The portrait characteristics combined matrix (Table 9) is obtained by vectors augmentation specified by tables 6–8:

$$G = \left(X \| Y \| \varphi\right). \quad (22)$$

Table 9 – G

№	G^X_i	G^Y_i	G^φ_i	№	G^X_i	G^Y_i	G^φ_i
1	-0.124	-0.22	0.806	9	-0.042	0.05	0.437
2	0.177	0.182	0.211	10	0.244	-0.028	0.578
3	-0.656	0.222	0.553	11	0.099	0.359	0.629
4	-0.224	-0.399	0.114	12	0.235	-0.044	0.504
5	0.088	0.343	0.752	13	0.072	0.619	0.696
6	0.347	0.017	0.543	14	-0.361	0.239	0.19
7	-0.015	-0.074	0.437	15	-0.075	-0.307	0.178
8	0.244	0.525	0.696	16	0.017	0.349	0.457

The shaded lines in table 9 correspond to points that did not fit in the unit square. In Fig. 6 the points defined by these rows of the matrix G are "masked" in the planar portrait.

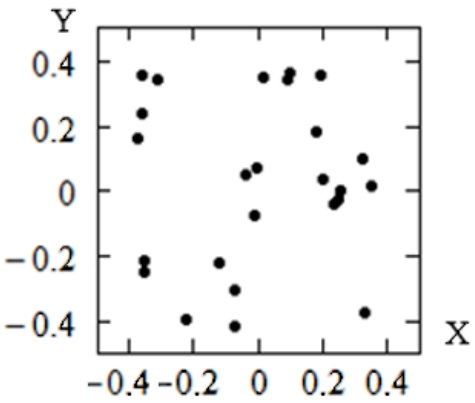


Fig. 6 – Planar portrait G

6 Example of portrait geometric center displacement implementation

To simulate the portrait geometric center displacement error defined by the matrix from table 9 we obtain a random vector of 2 elements using the sensor [0,1]:

$$Z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} 0.494 \\ 0.741 \end{pmatrix}. \quad (23)$$

Using the elements functional transformation (8) we proceed to the displacement values along the coordinates X and Y which distribution is subordinated to the centered trapezoidal function of the PDD (7):

$$d_X = d\left(z_1\right) = -0.002; \quad d_Y = d\left(z_2\right) = 0.096. \quad (24)$$

Then the following association will determine the displacement matrix:

$$Gd = \left(X + d_X \| Y + d_Y \| \varphi\right) = \left(Gd^X \| G\,d^Y \| Gd^\varphi\right). \quad (25)$$

The matrix and the corresponding portrait will have the form presented in table 10 and Fig. 7.

Table 10 – Gd

№	Gd^X_i	Gd^Y_i	Gd^φ_i	№	Gd^X_i	Gd^Y_i	Gd^φ_i
1	-0.126	-0.124	0.806	9	-0.044	0.146	0.437
2	0.175	0.279	0.211	10	0.242	0.068	0.578
3	-0.658	0.318	0.553	11	0.097	0.455	0.629
4	-0.226	-0.303	0.114	12	0.233	0.052	0.504
5	0.086	0.44	0.752	13	0.07	0.716	0.696
6	0.345	0.113	0.543	14	-0.363	0.335	0.19
7	-0.017	0.023	0.437	15	-0.077	-0.211	0.178
8	0.242	0.621	0.696	16	0.015	0.446	0.457

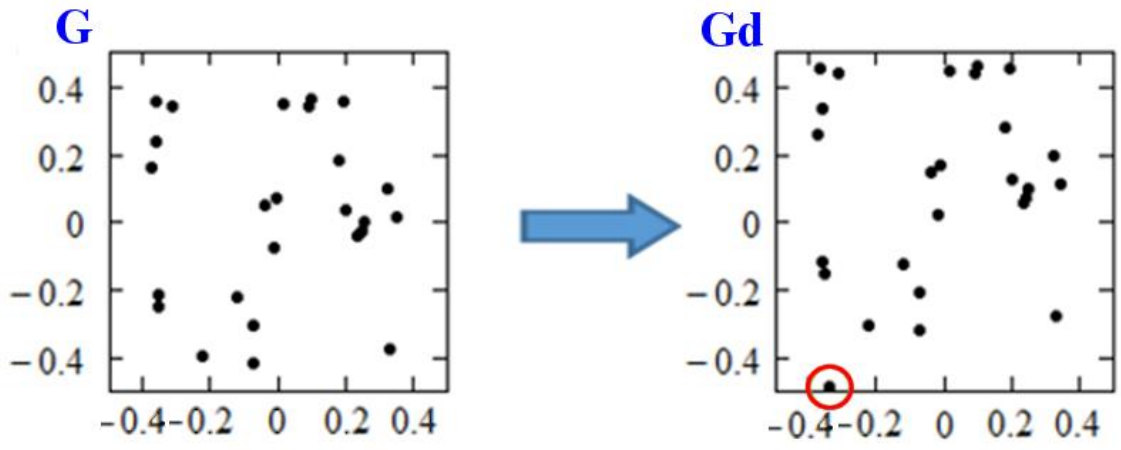


Fig. 7 – Original and displaced fingerprint portraits

However, the type of portrait is not final, because it is also can have the errors of random rotation and "erase-add".

7 Distorted portrait model converting

Make the following overrides:

- $G = Gctn$ - reference portrait is centered, truncated and normalized by the fingerprint rotation angle;
- $GD = Gd\alpha ea$ - portrait implementation distorted with random errors.

Matrix GDe consists the data about 27 portrait points.

Table 11 – GDe

№	$GDe^X{}_i$	$GDe^Y{}_i$	$GDe^\varphi{}_i$	№	$GDe^X{}_i$	$GDe^Y{}_i$	$GDe^\varphi{}_i$	№	$GDe^X{}_i$	$GDe^Y{}_i$	$GDe^\varphi{}_i$
1	-0.075	-0.161	0.863	11	-0.458	0.187	0.247	21	0.403	-0.147	0.682
2	0.066	0.322	0.268	12	0.002	-0.225	0.235	22	-0.449	0.301	0.623
3	-0.106	-0.363	0.171	13	-0.142	0.423	0.514	23	0.231	0.296	0.241
4	-0.074	0.442	0.809	14	0.199	0.178	0.155	24	-0.282	-0.267	0.612
5	0.283	0.227	0.6	15	-0.069	0.155	0.989	25	0.43	-0.494	0.3
6	-0.024	0.015	0.494	16	0.14	0.189	0.952	26	0.208	-0.387	0.722
7	-0.093	0.121	0.494	17	-0.441	0.106	0.468	27	-0.284	0.382	0.015
8	0.203	0.149	0.635	18	0.021	0.487	0.685	-	-	-	-
9	-0.069	0.46	0.686	19	0.042	-0.328	0.509	-	-	-	-
10	0.2	0.131	0.561	20	-0.297	-0.237	0.912	-	-	-	-

Centering portrait implementation

We calculate the mass center coordinates for the matrix GDe using the first two columns data:

$$\overline{Xe} = \frac{1}{Ne}\sum_{i=1}^{Ne} GDe^X{}_i \approx -0.016\,,\ \overline{Ye} = \frac{1}{Ne}\sum_{i=1}^{Ne} GDe^Y{}_i \approx 0.073\,. \qquad (26)$$

Conversion GDe by centering gives:

$$GDec = \left(GDe^X - \overline{Xe} \,\middle\|\, GDe^Y - \overline{Ye} \,\middle\|\, GDe^\varphi\right) = \left(GDec^X \,\middle\|\, GDec^Y \,\middle\|\, GDec^\varphi\right). \qquad (27)$$

Fig. 8 shows the obtained result by comparing the two portraits and we can see the resulting matrix in the Table 12.

One of the points left the unit normalized square limits and became one of the masked part.

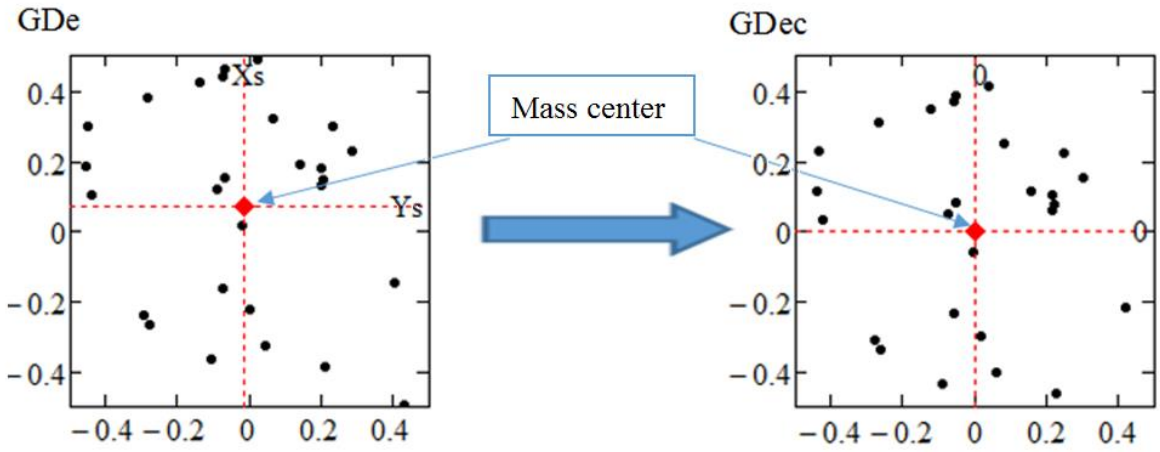


Fig. 8 – Transform portrait implementation after centering

Table 12 – Gdec

№	$GDec^X_i$	$GDec^Y_i$	$GDec^\varphi_i$	№	$GDec^X_i$	$GDec^Y_i$	$GDec^\varphi_i$	№	$GDec^X_i$	$GDec^Y_i$	$GDec^\varphi_i$
1	-0.059	-0.233	0.863	11	-0.441	0.114	0.247	21	0.419	-0.22	0.682
2	0.082	0.25	0.268	12	0.018	-0.297	0.235	22	-0.433	0.229	0.623
3	-0.09	-0.436	0.171	13	-0.126	0.35	0.514	23	0.247	0.224	0.241
4	-0.058	0.369	0.809	14	0.215	0.106	0.155	24	-0.266	-0.34	0.612
5	0.299	0.154	0.6	15	-0.053	0.082	0.989	25	0.446	-0.567	0.3
6	-0.008	-0.058	0.494	16	0.156	0.116	0.952	26	0.224	-0.46	0.722
7	-0.077	0.048	0.494	17	-0.425	0.033	0.468	27	-0.268	0.309	0.015
8	0.219	0.076	0.635	18	0.037	0.415	0.685	-	-	-	-
9	-0.053	0.387	0.686	19	0.058	-0.401	0.509	-	-	-	-
10	0.216	0.058	0.561	20	-0.281	-0.309	0.912	-	-	-	-

The truncation of points plurality in the portrait

We arrange the portrait matrix rows according to the points remoteness degree from the geometric center combined with the mass center. Calculate the points distance from the center

$$RD_i = \sqrt{\left(GDec_{i,0}\right)^2 + \left(GDec_{i,1}\right)^2}, \quad i = 1, \cdots, Ne. \tag{28}$$

Sorting rows according to values (28) gives the following matrix:

Table 13 – Gdecs

№	$GDecs^X_i$	$GDecs^Y_i$	$GDecs^\varphi_i$	№	$GDecs^X_i$	$GDecs^Y_i$	$GDecs^\varphi_i$	№	$GDecs^X_i$	$GDecs^Y_i$	$GDecs^\varphi_i$
1	-0.008	-0.058	0.494	11	0.247	0.224	0.241	21	-0.266	-0.34	0.612
2	-0.077	0.048	0.494	12	0.299	0.154	0.6	22	-0.09	-0.436	0.171
3	-0.053	0.082	0.989	13	-0.126	0.35	0.514	23	-0.441	0.114	0.247
4	0.156	0.116	0.952	14	-0.058	0.369	0.809	24	0.419	-0.22	0.682
5	0.216	0.058	0.561	15	-0.053	0.387	0.686	25	-0.433	0.229	0.623
6	0.219	0.076	0.635	16	0.058	-0.401	0.509	26	0.224	-0.46	0.722
7	0.215	0.106	0.155	17	-0.268	0.309	0.015	27	0.446	-0.567	0.3
8	-0.059	-0.233	0.863	18	0.037	0.415	0.685	-	-	-	-
9	0.082	0.25	0.268	19	-0.281	-0.309	0.912	-	-	-	-
10	0.018	-0.297	0.235	20	-0.425	0.033	0.468	-	-	-	-

Except points with numbers 17 – 27 from the range of lines.

Table 14 – Gdect

№	$Gect^X_i$	$Gect^Y_i$	$Gect^\varphi_i$	№	$Gect^X_i$	$Gect^Y_i$	$Gect^\varphi_i$	№	$Gect^X_i$	$Gect^Y_i$	$Gect^\varphi_i$
1	-0.008	-0.058	0.494	7	0.215	0.106	0.155	13	-0.126	0.35	0.514
2	-0.077	0.048	0.494	8	-0.059	-0.233	0.863	14	-0.058	0.369	0.809
3	-0.053	0.082	0.989	9	0.082	0.25	0.268	15	-0.053	0.387	0.686
4	0.156	0.116	0.952	10	0.018	-0.297	0.235	16	0.058	-0.401	0.509
5	0.216	0.058	0.561	11	0.247	0.224	0.241	-	-	-	-
6	0.219	0.076	0.635	12	0.299	0.154	0.6	-	-	-	-

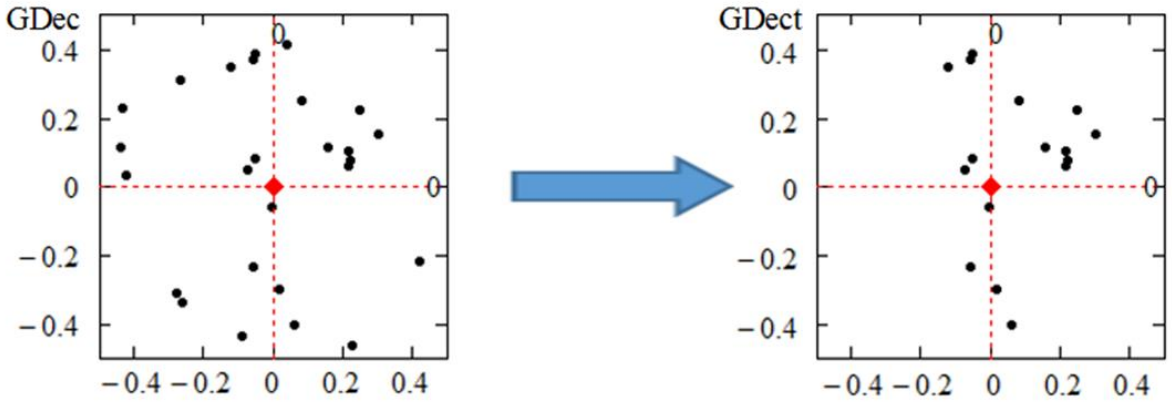


Fig. 9 – Portrait transformation after the points number limiting

Portrait position stabilization by the orientation angle on the plane

Make a portrait reversal GDect. Vector which indicates the points maximum concentration direction has to coincide with the axis O-X on the portrait plane. To do this, we define the transposed submatrix of matrix GDect:

$$XYect = \left(GDect^X, GDect^Y\right)^T. \tag{29}$$

Determine the distance of GDect portrait 16 points from the image geometric center:

$$r_i = \left|\left(XYect\right)_i\right|, \quad i = 1,\cdots,16 \tag{30}$$

and calculate the point projection coordinates matrix on the circle located around the center:

$$XYectr_i = \left(XYect\right)_i \cdot \left(2r_i\right)^{-1}, \quad i = 1,\cdots,16. \tag{31}$$

Direction vector of maximum concentration with points:

$$Vg = \sum_{i=1}^{16} XYectr_i = \sum_{i=1}^{16} \begin{pmatrix} XYectr_{i,1} \\ XYectr_{i,2} \end{pmatrix} = \begin{pmatrix} Vg_1 \\ Vg_2 \end{pmatrix} = \begin{pmatrix} 1.678 \\ 2.025 \end{pmatrix}. \tag{32}$$

Calculations (29) - (32) are illustrated in Fig. 10

The location of the vector Vg determines the required angle of axes rotation:

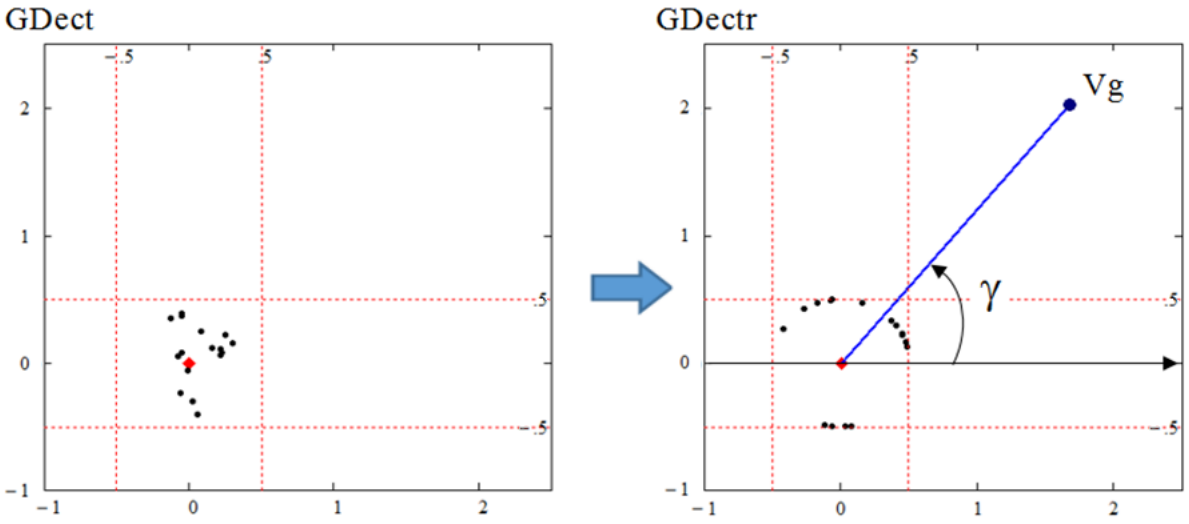
$$\gamma = \operatorname{arctg}\left(\frac{Vg_2}{Vg_1}\right) = 0.879\ [\text{рад}] = 50.355^o\ [\text{град}]. \tag{33}$$

When the axes rotate by an angle the submatrix of the stabilized portrait new coordinates is determined by the expression:

$$XYectn = \left[U(\gamma)\cdot\left(XYect\right)\right]^T. \tag{34}$$

Column vector modified with rotation arrival angles:

$$GDectn^{\varphi} = \left(GDect^{\varphi} - \frac{\gamma}{2 \cdot \pi} \right) \bmod 1 - \left\lfloor \left(GDect^{\varphi} - \frac{\gamma}{2 \cdot \pi} \right) \bmod 1 \right\rfloor . \quad (35)$$



а) centered truncated portrait;

б) the points projections location on the normalized circle and direction vector of maximum concentration with points.

Fig. 10 – Determination of the portrait angular stabilization parameters

Then the matrix of the fingerprint distorted implementation fully processed portrait will have a definition in the form:

$$GDectn = \left((XYectn) \| GDectn^{\varphi} \right). \quad (36)$$

The matrix contents are presented in table 15 and the portrait transformation with orientation angle stabilization on the plane is shown in Fig. 11.

Table 15 – Gdectn

№	$Gectn^{X}_{i}$	$Gectn^{Y}_{i}$	$Gectn^{\varphi}_{i}$	№	$Gectn^{X}_{i}$	$Gectn^{Y}_{i}$	$Gectn^{\varphi}_{i}$	№	$Gectn^{X}_{i}$	$Gectn^{Y}_{i}$	$Gectn^{\varphi}_{i}$
1	-0.05	-0.031	0.354	7	0.219	-0.098	0.015	13	0.189	0.32	0.375
2	-0.012	0.09	0.354	8	-0.217	-0.104	0.723	14	0.247	0.28	0.669
3	0.029	0.093	0.849	9	0.245	0.096	0.128	15	0.265	0.288	0.546
4	0.189	-0.046	0.812	10	-0.218	-0.203	0.096	16	-0.272	-0.301	0.369
5	0.182	-0.129	0.421	11	0.33	-0.047	0.101	-	-	-	-
6	0.198	-0.12	0.495	12	0.31	-0.132	0.461	-	-	-	-

The right image of fig. 11 represents the portrait final view processed according to the considered methods after the random errors action. The distorted portrait processing was carried out with the aim of distortion effect minimizing in order to approximate the characteristic points distribution to a reference sample similar distribution.

Fig. 12 shows a comparison of the distorted portrait processing result with the fingerprint obtained reference portrait normalized by the number of points and the spatial angle. Segments of straight lines indicate the distances of mutual displacement of the same points in the portraits. As we can see from the figure, in each of the portraits there is one point that does not have the corresponding pair (circled). This situation is explained by the results of the erasing and adding points processes, which are consequence of the corresponding errors.

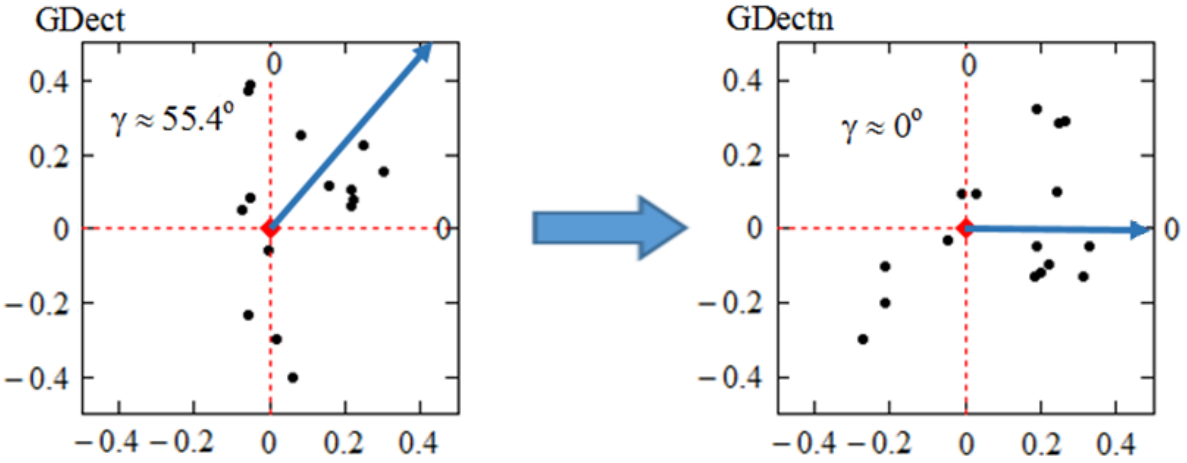


Fig. 11 – Fingerprint implementation rotation angle stabilization result

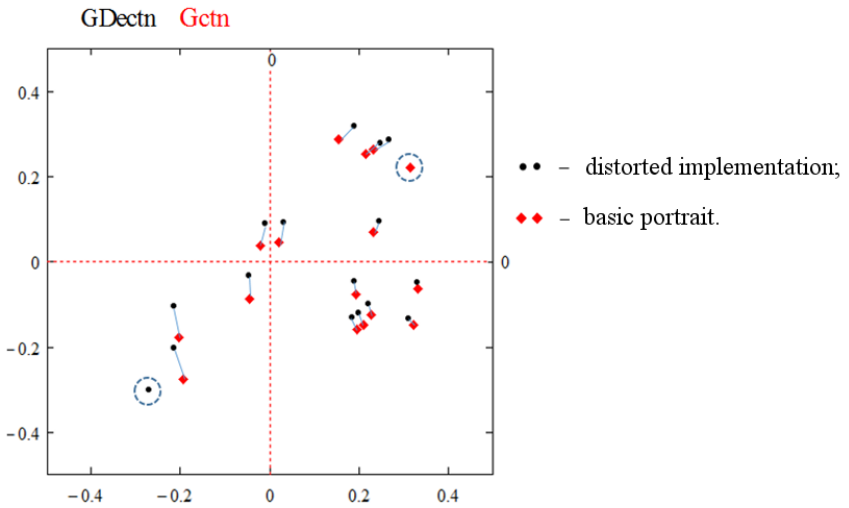


Fig. 12 – Comparison of the reference portrait points distribution with a similar distorted distribution after its processing

8 Conclusions

We have developed the mathematical models for the distorting fingerprints realizations process and for processing distorted images to minimize the random factors effect. As a result of processing the achieved value of the standard distance between the same points of the reference and processed distorted images was 4.8 % from the size of the normalized portrait square side. The proportion of points suitable for fingerprint identification by the location of the minutiae on the distorted implementation after it was processed by the proposed methods was 93.75 %.

The achieved result is characterized by a relatively high accuracy of the reference and repeated fingerprint portraits coincidence and is quite acceptable for identifying personalities based on correlation comparisons of the processed portrait with the biometric database samples. However, for the task of password access, when comparisons with standards are not permissible the indicators of high mean standard accuracy are not important. Effective secret-sharing algorithms are implemented without much difficulty if the introduced distortions do not lead to a change in the order of transferring the object characteristics (points) according to any ordered lexicographical rules. For example, from left to right and from top to bottom of the portrait square, in order of increasing distance from the center of the portrait , ascending arrival angles etc. It is obvious that even very small points displacements with respect to the reference image can cause a violation of any enumeration-listed orders.

References

[1] Rykanov A. S. Analys metodov raspoznavaniya otpechatkov pal'ca. Systemi obrabotki informacii. 2010. Vip.6. pp.164–171.
[2] Fan N. H., Spicyn V. G. Algoritmy dl'a klassifikacii otpechyatkov pal'cev na osnove primeneniya fil'tra Gabora, eyvlet-preobrazovania i mnogosloynoi neironnoy sety. Izvestia Tomskoho politechnicheskogo universiteta. Vol. 320, № 5: Upravlenie, vyichislitelnaya tehnika i informatika. pp. 60–64.
[3] Daktiloscopia.URL: http://www.ru.wikipedia.org
[4] Maltoni D., Maio D., Jain A.K., Prabhakar S. Handbook of fingerprint recognition. New York: Springer, 2003. 348 p.
[5] Zadorozhnyy V. Idenitifikacia po otpechatkam pal'cev. PC Magazine/Russian Edition. 2004. № 2. URL: http://www.bre.ru/security/20994.html
[6] Gasparyan A. V., Kyrakosyan A.A. Systema sravneniya otpechatkov pal'cev po localnym priznakam. Vestnyk RAU. Ser. fiziko-matematichaskiye i estestvennye nauki. 2006. Vip.2. pp. 85–91.
[7] Koleshko V. M., Vorobey E. A., Azizov P. M. Tradicionniye metodi biometricheskoy autentifikacii i identifikacii. Informatika cheloveka i biosistem: Uchebnoye elektronnoye izdaniye. Minsk: BNTU, 2009. 107 p.
[8] Gureeva O. Biometricheskaya identifikaciya po otpechatkam pal'cev. Tehnologiya Finger Chip. Komponentyi i tehnologii 2007. № 4. URL: http://www.kite.ru/articles/rfid/2007_4_176.php
[9] Bishop P. Atmel FingerChip Technology for Biometric Security. Atmel White Paper. URL: www.atmel.com
[10] Clarke R. Human Identification in Information Systems: Management Challenges and Public Policy Issues. Information Technology & People. 1994. Vol.7, Iss. 4. pp. 6–37 URL: https://doi.org/10.1108/09593849410076799

Рецензент: В'ячеслав Харченко, д.т.н., проф., академік Академії наук прикладної радіоелектроніки, Національний аерокосмічний університет ім. М. Є. Жуковського, м. Харків, Україна. E-mail: v_s_kharchenko@ukr.net

Надійшло: Червень 2018.

Автори:
Емілія Аніщенко, студентка кафедри Безпеки інформаційних систем і технологій, Харківський національний університет ім. В.Н. Каразіна, майдан Свободи 6, 61022, м. Харків, Україна. E-mail: emily9661@gmail.com
Сергій Рассомахін, д.т.н., зав. кафедри Безпеки інформаційних систем і технологій, Харківський національний університет ім. В. Н. Каразіна, майдан Свободи 6, 61022, м. Харків, Україна. E-mail: rassomakhin@karazin.ua

Математична модель і методи обробки біометричних зображень відбитків пальців.

Анотація. В даний час, використання технологій персональної ідентифікації на основі біометричних параметрів доступу до інформаційних ресурсів стає все більш актуальним у зв'язку з ростом інформатизації в сучасному суспільстві. Для підтвердження особи, використовуються такі фізичні характеристики, як обличчя, голос, сітківка ока або відбитки пальці. Найбільш вдалою технологією біометричної ідентифікації є порівняння відбитків пальців, завдяки простоті використання, відсутності стороннього втручання і надійності. У статті розглянута аналітична імовірнісна модель формування та обробки портретів відбитків пальців з урахуванням можливих природних факторів, здатних спотворити зображення відбитків, а також проведено деякі перетворення моделі, за допомогою яких можливо мінімізувати вплив спотворень для наближення виду розподілу характерних точок до аналогічного розподілу еталонного зразка.

Ключові слова: біометрія; модель; методи; відбитки пальців.

Рецензент: Вячеслав Харченко, д.т.н., проф., академик Академии наук прикладной радиоэлектроники, Национальный аэрокосмический университет им. М. Є. Жуковского, г. Харьков, Украина. E-mail: v_s_kharchenko@ukr.net

Поступила: Июнь 2018.

Авторы:
Емилия Анищенко, студентка кафедры Безопасности информационных систем и технологий Харьковский национальный университет им. В. Н. Каразина, пл. Свободы 6, 61022, г. Харьков, Украина. E-mail: emily9661@gmail.com
Сергей Рассомахин, д.т.н., зав. кафедры Безопасности информационных систем и технологий, Харьковский национальный университет им. В. Н. Каразина, пл. Свободы 6, 61022, г. Харьков, Украина. E-mail: rassomakhin@karazin.ua

Математическая модель и методы обработки биометрических изображений отпечатков пальцев.

Аннотация. В настоящее время, использование технологий персональной идентификации на основе биометрических параметров доступа к информационным ресурсам становится все более актуальным в связи с ростом информатизации в современном обществе. Для подтверждения личности, используются такие физические характеристики, как лицо, голос, сетчатка глаза или отпечатки пальце. Самой удачной технологией биометрической идентификации является сравнение отпечатков пальцев, благодаря простоте использования, отсутствию постороннего вмешательства и надежности. В статье рассмотрена аналитическая вероятностная модель формирования и обработки портретов отпечатков пальцев с учетом вероятных природных факторов, способных исказить изображение отпечатков, а также проведены некоторые преобразования модели, с помощью которых возможно минимизировать влияние искажений для приближения вида распределения характерных точек к аналогичному распределению эталонного образца.

Ключевые слова: биометрия; модель; методы; отпечатки пальцев.

UDC 004.056.55

ESSENCE AND CONDITIONS OF IMPLEMENTATION OF THE ATTACK BASED ON RELATED KEYS RELATIVELY ELECTRONIC SIGNATURES IBS-1 AND IBS-2 DSTU ISO/IEC 14888-3

Marina Yesina[1], Yuriy Gorbenko [1], Vladislav Kulibaba[1]

[1] V. N. Karazin Kharkiv National University, 6 Svobody Sq., Kharkiv, 61022, Ukraine
m.v.yesina@karazin.ua, gorbenkoU@iit.kharkov.ua, vlad.kulibaba1994@gmail.com

Reviewer: Roman Oliynikov, Doctor of Sciences (Engineering), Full Professor, V. N. Karazin Kharkiv National University, 6 Svobody Sq., Kharkiv, 61022, Ukraine
roliynykov@gmail.com

Received on July 2018

Abstract: *The paper deals with the state of protection electronic signatures based on the pairing of points of an elliptic curve against attacks based on the signing data with related keys. It is defined conditions and possibilities of the organization and implementation of these attacks. It is provided the recommendations on protection against these vulnerabilities, including in the post-quantum period.*

Keywords: *attack; electronic signature; elliptic curve; related keys; pairing.*

1 Introduction

Electronic signatures (ES) are now widespread, the stability of which is based on the complexity of discrete logarithm in finite fields and groups of elliptic curves (EC) points [3-5]. Also, researches were conducted and ES with appendix based on identity – the pairing of EC points are recommended for the application. Known conditions for implementation attacks based on related keys relative to ES based on standardized cryptographic transformations in finite fields and cyclic groups of supersingular curves. The conducted analysis of a large number of sources made it possible to conclude that there is no data regarding the security and conditions for implementation of attacks based on related keys with respect to ES IBS-1 and IBS-2 [1, 3-5], that are based on identity. At the same time, previous studies of the algorithms ES IBS-1 and IBS-2 stability showed that attack based on related keys can be implemented. Therefore, it is important to study the stability of these ES against attacks based on related keys.

Stability issues have become particularly relevant after statements and speeches of leading specialists about the potential vulnerabilities for the ES in the post-quantum period. Thus, the technical report of the US NSA [1] states that the ES, whose algorithms are based on transformation in the ring [1-2] and in the finite field [1-2], will be unstable with the appearance of quantum computers. The same suspicions are expressed in relation to cryptographic transformations in the group of elliptic curve points [1-2]. Therefore, the tasks and their solutions regarding the stability of the ES, which are now introduced in Ukraine and which operate on the international level, are important. Such standard should include DSTU ISO/IEC 14888-3:2014 [5].

One of the possible ways of solving this contradiction is to increase the size of the general parameters for the specified transformations. At the first stage of quantum cryptography development, this may work. But in the future it is necessary to apply other methods, for example, perhaps cryptographic transformation based on pairing of EC points and identification data. Such algorithms are offered in the DSTU ISO/IEC 14888-3:2014 in the form of algorithms ES IBS-1 and IBS-2 [5]. But the analysis showed that although they under certain conditions may qualify for post-quantum, further studies of their stability are needed. In our view, one of the vulnerabilities of the algorithms ES IBS-1 and IBS-2 is their vulnerability to attacks based on related keys.

The purpose of this article is to analyze the state of protection of the ES IBS-1 and IBS-2 against

the attacks based on the signed data with related keys, determine the conditions and possibilities for their organization and implementation, as well as the development of recommendations for the protection against the specified vulnerabilities, including post-quantum period.

2 The essence of ES IBS-1 and IBS-2, defined and implemented in DSTU ISO/IEC 14888-3

Given the novelty and the need to formulation of research problem of the ES IBS-1 and IBS-2, first we will consider the essence of these ES mechanisms and setup stages.

For the use of ES IBS-1 and IBS-2, at first, the general parameters must be entered and configured and asymmetric key pairs are generated.

The general parameters of the ES IBS-1 and IBS-2 are [3-5]:

- U – secret master key – integer, $U \in [1, q-1]$;
- V – public master key – EC point, $V = [U]P \bmod q$, $V \in G_1$;
- X – private (secret) signer key – EC point, $X = [U]Y \bmod q$, $X \in G_1$;
- Y – signer public key (verification) – EC point, $Y = H_1(ID) \bmod q$, $Y \in G_1$;
- P – base point of the key certification center of the order q.

General parameters generation or computation must be carried out under the following conditions:

- the user personal key X is calculated at his request in the key center generation (KCG) and provided to the user through a secure channel;
- the user public key Y can be calculated by each user of the domain;
- ID – is a data string that contains the signer identifier;
- H_1 – the hash function, which converts the data string into the element of group G_1;
- H_2 – the hash function defined in DSTU ISO/IEC 10118-3:2005;
- G_1 – a cyclic group of simple order q whose elements are EC points over $GF(p)$;
- G_2 – a cyclic group of simple order q whose elements are elements of a finite field $GF(p^m)$.

Tables 1 and 2 show the signing and verification mechanisms of IBS-1 and IBS-2 [5].

Table 1 – ES IBS-1 mechanism

Message signing	Signature verification
1) Generate a random or pseudo-random one-time secret key – an integer K, $1 < K < (q-1)$.	1) Verifier receives integral general parameters and subscriber public key.
2) Pairing: $\Pi =< X, P >^K$, $\Pi \in G_2$ over the field $GF(p^m)$, Π – pre-signature	2) Restoring one-time public key: – R and S recovered from the addition; – bit length R should be equal to the length of the function output H_2; – $S \in G_1$. If at least one of these conditions is not fulfilled, the signature is rejected.
3) Message in the form of an integer M is divided into its parts: M_2 – empty part, $M_1 = M$ – a message that needs to be signed.	3) Preparing to verify the message: – recovery M from signed message; – splitting message on M_1 & M_2: M_2 – empty, $M_1 = M$.
4) Calculating one-time public key: $R = H_2(M_1 \| FE2BS(\Pi))$, $R \in G_2$.	4) Restore parameter T: $T = (T_1, T_2)$, $T_1 = -Y$, $T_2 = [R]Y$.

Continuation of Table 1

Message signing	Signature verification
5) Calculate parameter T: $T=(T_1,T_2)=(-Y,[R]Y)$.	5) Pairing: $\overline{\Pi}=<S,P>\times<Y,V>^R$.
6) Calculate signature component: $S=[K-R]X \bmod q$, $S\in G_1$. Signature is $\Sigma=(R,S)$.	6) One-time validation public key calculation: $\overline{R}=H_2(M_1 \| FE2BS(\overline{\Pi}))$.
7) Construction of the addition with the concatenation of the text in the form $(R,S)\|text$.	7) Comparison $\overline{R}=R$: if they do not match, the signature is false, otherwise – true.
8) Building the signed message in the form $M((R,S)\|text)$.	

Table 2 – ES IBS-2 mechanism

Message signing	Signature verification
1) Generate a random or pseudo-random one-time secret key – an integer K, $1<K<(q-1)$.	1) Verifier receives integral valid general parameters and valid subscriber public key.
2) Scalar Multiplication: $\Pi=[K]Y \bmod q$, $\Pi\in G_1$, Π – pre-signature, EC point.	2) Restoring one-time public key: – R and S recovered from the addition; – $R\in G_1$, $S\in G_1$. If at least one of these conditions is not fulfilled, the signature is rejected.
3) Message in the form of an integer M is divided into its parts: M_1 – empty part, $M_2=M$ – a message that needs to be signed.	3) Preparing to verify the message: – recovery M from signed message; – splitting message on M_1 & M_2: M_1 – empty, $M_2=M$.
4) Calculating one-time public key $R=\Pi$, $R\in G_1$.	4) Restore parameter T: $T=(T_1,T_2)$, $T_1=-Y$, $T_2=[-H]Y$, $H=H_2(M_2 \| FE2BS(R_x))$.
5) Calculate parameter T: $T=(T_1,T_2)=(-Y,[-H]Y)$, $H\in G_2$, $H=H_2(M_2 \| FE2BS(\Pi_x))$.	5) Calculating pre-signature: $\overline{\Pi}=R$, $\overline{\Pi}\in G_1$.
6) Calculate signature component: $S=[K+H]X \bmod q$, $S\in G_1$. Signature is $\Sigma=(R,S)$.	6) Calculating: $\overline{R}_1=<P,S>$ та $\overline{R}_2=<V,\overline{\Pi}+[H]Y>$.
7) Building the addition: $(R,S)\|text$.	7) Comparison $\overline{R}_1=\overline{R}_2$: if they do not match, the signature is false, otherwise – true.
8) Building the signed message: $M((R,S)\|text)$.	

3 Attack "Full Disclosure" against ES IBS-1 based on signed data and related keys

Let the cryptanalyst intercepts and has a full access to i signed messages [2,5]:

$$\begin{cases} S_1=[K_1-R_1]X \bmod q \\ ..... \\ S_i=[K_i-R_i]X \bmod q \end{cases}. \quad (1)$$

The system (1) includes i equations and $i+1$ unknowns.

Find an unknown EC point – a private long-term key X, which is permanent for all signatures. As a result, we obtain the system of the form:

$$\begin{cases} X = [K_1 - R_1]^{-1} S_1 \bmod q \\ \dots \\ X = [K_i - R_i]^{-1} S_i \bmod q \end{cases}. \quad (2)$$

In system (2) private long-term key X is unknown and i unknowns $K_1, K_2, ..., K_i$. For full disclosure, that is, the definition of a secret key X by i ES, it is necessary to solve the system of the i-th order with the $i+1$ unknowns. The analysis showed that it is practically impossible to reduce the system of equations order by the force method. Therefore, we can assume that the attack based on signed data has an exponential complexity [2, 5].

As the analysis showed, one of the possible variants of lowering the system of equations order can be the key related, for example, in the form [2]:

$$K_1 + K_2 = q \quad (3)$$

or otherwise. Consider an attack based on related keys.

We write the system (1) for the case of two equations, and consider the signature algorithms for the two messages M_1 and M_2, and keys that satisfy the condition (3).

For message M_1	For message M_2
$K_1 \in [1, q-1]$	$K_2 = (q - K_1) \in [1, q-1]$
$\Pi_1 =< X, P >^{K_1}$	$\Pi_2 =< X, P >^{K_2}$
$R_1 = H_2(M_1 \| FE2BS(\Pi_1))$	$R_2 = H_2(M_2 \| FE2BS(\Pi_2))$
$S_1 = [K_1 - R_1]X \bmod q$	$S_2 = [(q - K_1) - R_2]X \bmod q$

Next we find a condition in which $S_1 = S_2$, that is, we will find a personal key X, in which ES of messages M_1 and M_2 coincide. As a result, we have:

$$[K_1 - R_1]X \bmod q = [(q - K_1) - R_2]X \bmod q. \quad (4)$$

We reduce in (4) by X, as a result we obtain:

$$[K_1 - R_1] \bmod q = [(q - K_1) - R_2] \bmod q; \quad (5)$$

$$[K_1 - R_1] \bmod q = [-K_1 - R_2] \bmod q; \quad (6)$$

$$2K_1 \bmod q = [R_1 - R_2] \bmod q. \quad (7)$$

Next we will find from (7) a one-time key K_1, since R_1 and R_2 are known and are contained in the signature:

$$K_1 = \frac{R_1 - R_2}{2} \bmod q. \quad (8)$$

Thus, the system of equations order is reduced to an unknown one-time secret key, in our case K_1:

$$\begin{cases} X = [K_1 - R_1]^{-1} S_1 \bmod q \\ \dots \\ X = [K_i - R_i]^{-1} S_i \bmod q \end{cases}. \quad (9)$$

Substituting K_1, and in general K_j, into system (9), we have a system of i equations with i unknowns, which has a solution.

4 Attack "Full Disclosure" against ES IBS-2 based on signed data and related keys

Analogously to (1), for IBS-2, taking into account Table 2, we have [2,5]:

$$\begin{cases} S_1 = [K_1 + H_1]X \bmod q \\ ..... \\ S_i = [K_i + H_i]X \bmod q \end{cases}. \quad (10)$$

Next, we will find from (10) a private long-term key X and we will receive the following for it:

$$\begin{cases} X = [K_1 + H_1]^{-1} S_1 \bmod q \\ ... \\ X = [K_i + H_i]^{-1} S_i \bmod q \end{cases}. \quad (11)$$

System (10) includes i equations and $i+1$ unknowns in receiving i signed messages. The main task of the cryptanalysis is to identify the private long-term key X.

As in the case (2), as shown by the analysis, it is practically impossible to reduce the system of equations (11) by force. Moreover, the complexity of a force attack is determined by the order of the cyclic group q. Therefore, we can assume that the complexity of the attack based on the signed data is exponential [2,5].

At the same time, as in the case (2), one of the possible options of the system of equations (11) reduction may be the key related, for example, in the form (3) or another way [2].

We write the system (11) for the case of two equations and specified key related, and consider the signature algorithms for the two messages M_1 and M_2, and keys that satisfy the condition (3).

For message M_1	For message M_2
$K_1 \in [1, q-1]$	$K_2 = (q - K_1) \in [1, q-1]$
$\Pi_1 = [K_1]Y \bmod q$	$\Pi_2 = [K_2]Y \bmod q =$
	$= [q - K_1]Y \bmod q =$
	$= [-K_1]Y \bmod q$
$R_1 = \Pi_1$	$R_2 = \Pi_2$
$S_1 = [K_1 + H_1]X \bmod q$	$S_2 = [K_2 + H_2]X \bmod q =$
	$= [(q - K_1) + H_2]X \bmod q =$
	$= [-K_1 + H_2]X \bmod q$

We find a condition in which $S_1 = S_2$. As a result, we have

$$[K_1 + H_1]X \bmod q = [(q - K_1) + H_2]X \bmod q. \quad (12)$$

Reduce (12) at X, we obtain that:

$$[K_1 + H_1] \bmod q = [(q - K_1) + H_2] \bmod q$$

or

$$[K_1 + H_1] \bmod q = [-K_1 + H_2] \bmod q$$

and

$$2K_1 \bmod q = [H_2 - H_1] \bmod q. \quad (13)$$

Finally, from (13) we obtain that

$$K_1 = \frac{H_2 - H_1}{2} \bmod q. \quad (14)$$

Thus, the system of equations (11) order is reduced, since the values of H_1 and H_2 can be used to define an unknown one-time secret key K_1.

5 An example of the "Full Disclosure" attack against the mechanisms ES IBS-1 and IBS-2 based on signed data and related keys

Let's show the correctness of the execution of attacks on the example. Determine the value of the required parameters – the value of the base point P, the user private key X and the order of the base point q: $X = (13,16)$, $P = (13,7)$, $q = 7$. EC over the main field: $y^2 = (x^3 + x + 1) \bmod 23$.

Consider an example for the mechanism IBS-1 [3-5].

We write the system (1) for the case of two equations and consider the signature algorithms for the two messages M_1 and M_2, and keys that satisfy the condition (3).

For message M_1	For message M_2
$K_1 = 6$	$K_2 = (q - K_1) = 1$
$\Pi_1 =< X, P >^{K_1}$	$\Pi_2 =< X, P >^{K_2}$
$R_1 = H_2(M_1 \| FE2BS(\Pi_1)); R_1 = 4$	$R_2 = H_2(M_2 \| FE2BS(\Pi_2)); R_2 = 20$
$S_1 = [K_1 - R_1]X \bmod q$	$S_2 = [K_2 - R_2]X \bmod q$
$S_1 = [6-4](13,16) \bmod 7 =$	$S_2 = [1-20](13,16) \bmod 7 =$
$= 2(13,16) \bmod 7 = (5,19)$	$= (-19)(13,16) \bmod 7 = (5,19)$

According to formula (8) for K_1 we obtain:

$$K_1 = \frac{4-20}{2} \bmod 7 = \frac{-16}{2} \bmod 7 = (-8) \bmod 7 = 6.$$

Let's solve the equation from the system (2), by substituting the obtained value K_1:

$$X = [K_1 - R_1]^{-1} S_1 \bmod q;$$

$$X = [6-4]^{-1}(5,19) \bmod 7 = (2)^{-1}(5,19) \bmod 7.$$

Find the inverse element in the field:

$$z = \frac{1}{k} \bmod q; \; z \cdot k = 1 \bmod q; z \cdot 2 = 1 \bmod 7; \; z = 4.$$

So, after finding the inverse element, we obtain the following:

$$X = 4(5,19) \bmod 7 = (13,16).$$

We found the value of the signer private key X. Compare it with X:

$$X = (13,16), X = (13,16) \Rightarrow X = X.$$

Consequently, for mechanism IBS-1, the considering attack is realizing.

Consider an example for the mechanism IBS-2 [3-5].

We write the system (10) for the case of two equations and consider the signature algorithms for the two messages M_1 and M_2, and keys that satisfy the condition (3).

For message M_1	For message M_2
$K_1 = 6$	$K_2 = (q - K_1) = 1$
$\Pi_1 = [K_1]Y \bmod q$	$\Pi_2 = [-K_1]Y \bmod q$
$\Pi_1 = 6(17,20) \bmod 7 =$	$\Pi_2 = -6(17,20) \bmod 7 =$
$= (17,3)$	$= (17,20)$
$R_1 = \Pi_1 = (17,3)$	$R_2 = \Pi_2 = (17,20)$
$S_1 = [K_1 + H_1]X \bmod q \quad H_1 = 3$	$S_2 = [(q - K_1) - R_2]X \bmod q \quad H_2 = 15$
$S_1 = [6+3](13,16) \bmod 7 =$	$S_2 = [-6+15](13,16) \bmod 7 =$
$= 2(13,16) \bmod 7 = (5,19)$	$= 2(13,16) = (5,19)$

According to formula (14) for K_1 we obtain:

$$K_1 = \frac{15-3}{2} \bmod 7 = \frac{12}{2} \bmod 7 = 6.$$

Let's solve the equation from the system (11), by substituting the obtained value K_1:

$$X = [K_1 + H_1]^{-1} S_1 \bmod q;$$

$$X = [6+3]^{-1}(5,19) \bmod 7 = (9)^{-1}(5,19) \bmod 7.$$

Find the inverse element in the field:

$$z = \frac{1}{k} \bmod q; \; z \cdot k = 1 \bmod q; \; z \cdot 9 = 1 \bmod 7; \; z = 4.$$

So, after finding the inverse element, we obtain the following:

$$X = 4(5,19) \bmod 7 = (13,16).$$

We found the value of the signer private key X. Compare it with X:

$$X = (13,16), X = (13,16) \Rightarrow X = X.$$

Consequently, for mechanism IBS-2, the considering attack is realizing.

6 An example of the "Full Disclosure" attack against the mechanisms ES IBS-1 and IBS-2 based on signed data and related keys: alternative approach

Let the cryptanalyst intercepts and has a full access to i signed messages: similar to (1) and (10).

Find an unknown point of the EC – a private long-term key X, which is permanent for all signatures. Consider an attack on the ES IBS-1 based on the key related. Input data will be the similar to those given in section 3 [5].

As a result, we obtain for IBS-1 the following system:

$$\begin{cases} S_1 = [K_1 - R_1]X \bmod q \\ S_2 = [-K_1 - R_2]X \bmod q \end{cases}$$
$$S_1 + S_2 = [(K_1 - R_1) + (-K_1 - R_2)]X \bmod q$$
$$S_1 + S_2 = [-R_1 - R_2]X \bmod q \quad . \qquad (15)$$
$$X = (S_1 + S_2)[-R_1 - R_2]^{-1} \bmod q$$
$$X = -[R_1 + R_2]^{-1}(S_1 + S_2) \bmod q$$

Consider an attack on the ES IBS-2 based on the key related. Input data will be similar to the data given in section 4 [5].

As a result, we obtain for IBS-2 the following system:

$$\begin{cases} S_1 = [K_1 + H_1]X \bmod q \\ S_2 = [-K_1 + H_2]X \bmod q \end{cases}$$

$$S_1 + S_2 = [(K_1 + H_1) + (-K_1 + H_2)]X \bmod q . \quad (16)$$

$$S_1 + S_2 = [H_1 + H_2]X \bmod q$$

$$X = [H_1 + H_2]^{-1}(\mathrm{S}_1 + \mathrm{S}_2) \bmod q$$

Now let's give a mathematical example and correctness of attacks execution on an example.

First, consider an example for the mechanism IBS-1. Input data will be similar to the data given in section 5.

Using system (15), to find the signer private key X, we have the following:

$$\mathrm{X} = -[\mathrm{R}_1 + \mathrm{R}_2]^{-1}(\mathrm{S}_1 + \mathrm{S}_2) \bmod q ;$$

$$\mathrm{X} = -[4 + 20]^{-1}((5,19) + (5,19)) \bmod 7 ;$$

$$\mathrm{X} = (-24)^{-1}(2(5,19)) \bmod 7 .$$

Find the inverse element in the field:

$$z = \frac{1}{k} \bmod q ; z \cdot k = 1 \bmod q ;$$

$$z \cdot (-24) = 1 \bmod 7 ; \ z = 2 .$$

So, after finding the inverse element, we obtain the following:

$$X = 2(17,3) \bmod 7 = (13,16) ; \quad (17)$$

$$X = (13,16), X = (13,16) \Rightarrow X = X .$$

So, according to (17), for mechanism IBS-1, the considering attack is realizing.

Let's consider an example for the mechanism IBS-2. Input data will be similar to the data given in section 5.

Using system (16), to find the signer private key X, we have the following:

$$\mathrm{X} = [H_1 + H_2]^{-1}(\mathrm{S}_1 + \mathrm{S}_2) \bmod q ;$$

$$\mathrm{X} = [3 + 15]^{-1}((5,19) + (5,19)) \bmod 7 ;$$

$$\mathrm{X} = [4]^{-1}(17,3) \bmod 7 .$$

Find the inverse element in the field:

$$z = \frac{1}{k} \bmod q ; z \cdot k = 1 \bmod q ; z \cdot 4 = 1 \bmod 7 ; \ z = 2 .$$

So, after finding the inverse element, we obtain the following:

$$\mathrm{X} = 2(17,3) \bmod 7 = (13,16) ; \quad (18)$$

$$X = (13,16), X = (13,16) \Rightarrow X = X .$$

So, according to (18), for mechanism IBS-2, the considering attack can be implemented with polynomial complexity.

7 Proposals for ES IBS-1 and IBS-2 algorithms protection against attacks based on related keys

The analysis allows to propose the following mechanisms for protecting the ES IBS-1 and IBS-2 from attacks based on related keys [2,5].

1. Based on encryption of signed messages using symmetric or asymmetric ciphers. In terms of the complexity (*performance*) of encryption and stability, it is better to use symmetric ciphers–block or stream. Then cryptanalysis will need to solve the system with $2i+1$ unknowns, but for a system with i equations. Such task is exponentially complicated with real parameter values.

2. Another mechanism for protecting ES IBS-1 and IBS-2 against attacks based on related keys is the exclusion of the ability to relate one-time keys K in the process of signing a message flow. This can be done on the basis of the use of hardware or hardware-software ES means, which would exclude the possibility of interference in the process of signing messages. Other mechanisms of ES are also possible.

8 Conclusions and recommendations

1. In the process of ES improving, ES algorithms IBS-1 and IBS-2 on identifiers with pairing of EC points are proposed, in them as a private key is proposed to use the elliptic curve point X. As a result, when intercepting i signed messages to determine the long-term key X, it is necessary to solve a system of equations with $i+1$ unknown, i of which there are large random numbers, one is the EC point X. In the process of analysis, no effective methods have been identified for the solution of such system.

2. It was discovered that the ES IBS-1 and IBS-2 cryptographic transformation do not provide cryptographic resistance against attacks based on related key. At that, two different variants of attack based on related key were obtained.

3. For ES IBS-1 algorithm, attack based on related key can be accomplished using the obtained relations (8) and (9). Moreover, its complexity is polynomial.

4. For ES IBS-2 algorithm, attack based on related key can be accomplished using the obtained relations (11) and (14). Its complexity is also polynomial.

5. The possibility of attacks against ES algorithms IBS-1 and IBS-2 is confirmed not only by software simulation, but also by the examples given in section 5 of this paper.

6. Another method of attacking ES algorithms IBS-1 and IBS-, the essence of which is described in section 6 of this paper, in particular using systems (15) and (16), is also revealed. The above attacks also have a polynomial complexity. The ability to perform these attacks is also demonstrated in the examples.

7. Thus, both theoretically and in the examples, it is shown that the ES algorithms IBS-1 and IBS-2 are unstable against attacks based on related key, so if they are used, they must use protection mechanisms against such attacks.

8. Suggestions on possible options for protecting the ES for algorithms DSTU ISO/IEC 14888-3:2014 – IBS-1 and IBS-2 against attacks based on related key are also outlined above. The main ones are the encryption of signed messages and the use of qualified hardware and software ES.

References

[1] Koblitz N., Menezes A.J. A riddle wrapped in an enigma. URL: https://eprint.iacr.org/2015/1018.pdf.

[2] Gorbenko I.D., Gorbenko Yu.I. Prykladna kryptologija. Teorija. Praktyka. Zastosuvannja: monografija. Harkiv: Fort, 2012. 870 p.

[3] Gorbenko Yu.I., Ganzja R.S., Akol'zina O.S. Elektronni pidpysy na osnovi identyfikatoriv ta binarnogo vidobrazhennja. Prikladnaya radioelektronika. 2015. T. 14, № 4. pp. 284–290.

[4] Gorbenko Yu.I., Jesina M.V., Kulibaba V.A. Sutnist' ta umovy zdijsnennja ataky na zv'jazanyh kljuchah vidnosno elektronnyh pidpysiv IBS-1 ta IBS-2 DSTU ISO/IEC 14888-3. Systemy obrobky informacii'. 2016. № 7(144). pp. 113–118.

[5] DSTU ISO/IEC 14888-3:2014 Informacijni tehnologii'. Metody zahystu. Cyfrovi pidpysy z dopovnennjam. Chast.3. Mehanizmy, shho g'runtujut'sja na dyskretnomu logaryfmi (ISO/IEC 14888-3:2008, IDT). 2014. 113 p.

Рецензент: Роман Олійников, д.т.н., проф., Харківський національний університет імені В.Н. Каразіна, майдан Свободи 6, м. Харків, 61022, Україна.
E-mail: roliynykov@gmail.com

Надійшло: Липень 2018.

Автори:
Марина Єсіна, к.т.н., старший викладач кафедри безпеки інформаційних систем і технологій, Харківський національний університет імені В.Н. Каразіна, майдан Свободи 4, м. Харків, 61022, Україна.
E-mail: m.v.yesina@karazin.ua

Юрій Горбенко, к.т.н., провідний науковий співробітник, Харківський національний університет імені В.Н. Каразіна, майдан Свободи 4, м. Харків, 61022, Україна.
E-mail: gorbenkoU@iit.kharkov.ua

Владислав Кулібаба, аспірант кафедри безпеки інформаційних систем і технологій, Харківський національний університет імені В.Н. Каразіна, майдан Свободи 4, м. Харків, 61022, Україна.
E-mail: vlad.kulibaba1994@gmail.com

Сутність та умови здійснення атаки на зв'язаних ключах відносно електронних підписів IBS-1 та IBS-2 ДСТУ ISO/IEC 14888-3.

Анотація. У роботі розглядається стан захищеності електронних підписів на основі спарювання точок еліптичної кривої від атак на основі підписаних даних зі зв'язаними ключами. Визначаються умови та можливості організації та реалізації цих атак. Надаються рекомендації відносно захисту від вказаних вразливостей, в тому числі у постквантовий період.

Ключові слова: атака; електронний підпис; еліптична крива; зв'язані ключі; спарювання.

Рецензент: Роман Олейников, д.т.н., проф., Харьковский национальный университет имени В.Н. Каразина, пл. Свободы 6, г. Харьков, 61022, Украина.
E-mail: roliynykov@gmail.com

Поступила: Июль 2018.

Авторы:
Марина Есина, к.т.н., старший преподаватель кафедры безопасности информационных систем и технологий, Харьковский национальный университет имени В. Н. Каразина, пл. Свободы 6, г. Харьков, 61022, Украина.
E-mail: m.v.yesina@karazin.ua

Юрий Горбенко, к.т.н., ведущий научный сотрудник, Харьковский национальный университет имени В. Н. Каразина, пл. Свободы 6, г. Харьков, 61022, Украина.
E-mail: gorbenkoU@iit.kharkov.ua

Владислав Кулибаба, аспирант кафедры безопасности информационных систем и технологий, Харьковский национальный университет имени В. Н. Каразина, пл. Свободы 6, г. Харьков, 61022, Украина.
E-mail: vlad.kulibaba1994@gmail.com

Сущность и условия выполнения атаки на связанных ключах относительно электронных подписей IBS-1 и IBS-2 ДСТУ ISO/IEC 14888-3.

Аннотация. В работе рассматривается состояние защищенности электронных подписей на основе спаривания точек эллиптической кривой от атак на основе подписанных данных со связанными ключами. Определяются условия и возможности организации и реализации этих атак. Предоставляются рекомендации относительно защиты от указанных уязвимостей, в том числе в пост квантовый период.

Ключевые слова: атака; электронная подпись; эллиптическая кривая; связанные ключи; спаривание.

UDC 004.056.55

CODE BASED FUZZY EXTRACTOR FOR BIOMETRIC KEYS

Alexandr Kuznetsov[1], Anastasia Kiyan[1], Roman Serhiienko[2], Anna Uvarova[3], Dmytro Prokopovych-Tkachenko[4]

[1] V. N. Karazin Kharkiv National University, 6 Svobody Sq., Kharkiv, 61022, Ukraine
kuznetsov@karazin.ua, nastyak931@gmail.com,

[2] National Army Academy named after Hetman Petro Sahaidachnyi, 32 Heroes of Maidan St., Lviv, 79012, Ukraine
romanserg69@gmail.com

[3] Yuzhnoye State Design Office, Dnipro, 3 Krivorozhskaya St., 49008, Ukraine
annet.uvarova@gmail.com

[4] University of Customs and Finance, 2/4 Volodymyr Vernadsky St., Dnipro, 49000, Ukraine
omega2@email.dp.ua

Reviewer: Sergii Kavun, Doctor of Sciences (Economics), Ph.D. (Engineering), Full Prof., Kharkiv University of Technology "STEP", Kharkiv, 61010, Ukraine
kavserg@gmail.com

Received on August 2018

Abstract. *In this paper methods of forming cryptographic keys from biometric images using fuzzy extractors are considered. A new scheme of a fuzzy extractor based on the McEliece cryptosystem is proposed. It is shown that the new design of the fuzzy extractor allows forming cryptographic passwords from biometric images even without the use of non-secret helper string. When using helper string, the proportion of corrected distortions of biometric images increases significantly. In addition, the proposed design relates to a class of post-quantum information security methods, i.e. it is expected to be safely used even for solving cryptanalysis problems with universal quantum computers.*

Keywords: *code based cryptosystem; fuzzy extractor; biometric cryptography; cryptographic keys.*

1 Introduction

Biometric authentication methods [1-12] are an important area of modern research in the field of cyber security. They are widely used in various applications: criminology, e-commerce, copyright protection, electronic document management, and so on.

In recent years, interest in biometric methods has considerably expanded. From traditional biometric systems based on the comparison of biometric images with stored reference copies modern technologies have switched to the formation of cryptographic keys "on the fly." In this case, biometric data no longer requires the storage, transmission, complex and costly means of protection etc., the possibility of their intentional and/or accidental compromise is excluded. All verification, identification and authentication procedures are performed using depersonalized cryptographic keys (*passwords, access codes, PIN-codes*), and the unique biometric personal data remains safe. These formed depersonalized key sequences further will be called "biometric keys".

The next step in the development of such technologies will be the creation of high-quality biometric cryptographic systems in which biometric personal data should be used as a source of unique secret parameters. At the same time, the user will not need to memorize cryptographic keys (*passwords*) and / or use additional devices for their storage, transmission and etc. The biometric cryptosystem is initialized at any time and in any place by extracting "on the fly" the required parameters from the provided biometric images (with possible errors, erasures, etc.) without compromising these images. At the same time it is necessary to provide the maximum range of services and safety guarantees, taking into account the peculiarities of the construction of biometric cryptosystems.

In this paper, we consider methods of forming cryptographic keys from biometric images[1] using fuzzy extractors [3,4].

[1] The term "biometric images (data)" hereinafter refers to sets of biometric characteristics that can be represented in the form of binary vectors that can be compared in the Hamming metric. It is assumed that the different sets of characteristics of the same user differ from each other by no more than 25% (this threshold corresponds to the limiting corrective capabilities of noise immunity codes).

Traditionally, fuzzy extractors, as well as fuzzy containers [2] preceding them, are constructed using the methods of noise immune coding. At the initial stage, biometric data in some sense are "united" with elements of noise-immune codes (for example, codewords or syndrome sequences). For fuzzy extractors an additional helper string is created which "helps" in extracting the secret parameter based on fuzzy biometric set. In the direct use phase noise immune decoding is used, which eliminates the possibility of uncertainty (caused by distortions, erasures, etc.) in user-provided biometric images. If the differences in the sets of characteristics are not large (don't exceed the error-correcting ability of the code), then fuzzy extractors (vaults) allow uniquely to restore the secret parameter (*biometric key*).

In this paper a new scheme of a fuzzy extractor based on the McEliece cryptosystem [13] is proposed. It is shown that the new design allows generation cryptographic passwords from biometric images even without non-secret helper string. When using helper string, the proportion of corrected distortions of biometric images increases significantly. Besides, the proposed design relates to a class of post-quantum information security methods, i.e. it is expected to be safely used even for solving cryptanalysis problems with universal quantum computers.

2 Fuzzy Vault and Fuzzy Extractor

In [1] the forming of a secret key using biometrics is considered, the simplified scheme of which is shown in fig. 1. At the initial stage a secret parameter (key) K is generated, which is encoded by a noise immune code. Biometric user data B is added to the received codeword. The resulting "carrier" $B+c$ is actually a noise-masked biometric secret key. If biometrics B^* which is close to the original data $B^* \approx B$ at the usage stage would be provided, then after subtraction of B decoding will restore the secret key.

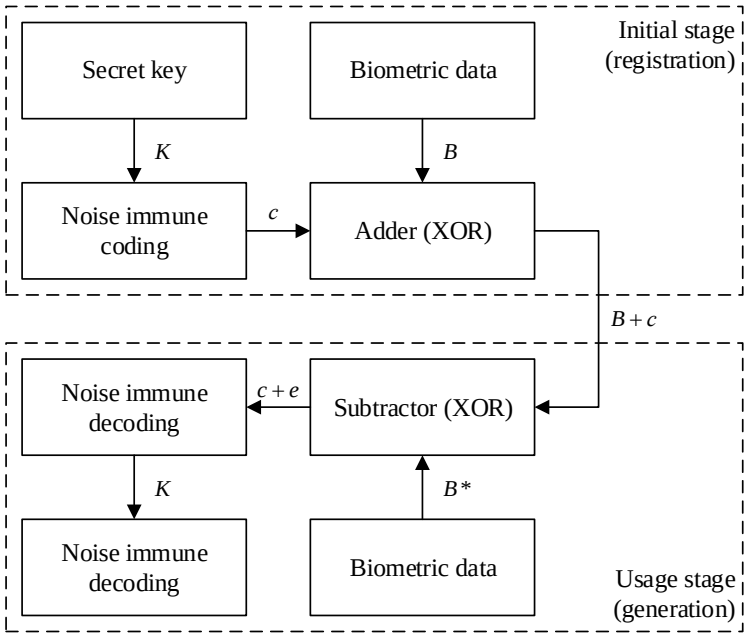


Fig. 1 – Scheme of a biometric key forming

Indeed, after subtraction we obtain: $(B+c)-B^*=c+e$, where $e=B-B^*$ is considered as vector of errors.

If the Hamming weight of the vector e (the number of its non-zero components) does not exceed the corrective capacity of the noise immune code t, then the decoding of the vector $(B+c)-B^*$ will allow to find the vector c, vector e and, as a consequence, the key parameter K.

Obviously, the cryptographic properties of the scheme [1] depend on both the selected noise immune code and method of forming biometric data. The encoded secret parameter K is contained in the "carrier" $B+c$ and, obviously, there are possible statistical attacks that recover the codeword c and the secret key K.

The scheme of the fuzzy storage was first proposed in [2]. It is also based on the use of noise immune codes. The secret parameter is "hidden" in the encoded set of data provided by user. Any user will be able to extract the secret parameter only if his set is close to the original set, and minor differences will be corrected in the process of noise immune decoding. A statistical analysis of a fuzzy vault likely could lead to a possible attack on the stored secret key.

The biometric keys technologies further were developed in [3-12] etc. In particular, in the fundamental papers [3,4] so-called fuzzy extractors where proposed, whose designs are very close to the keys forming schemes from [1]. The main ones are two constructions [3,4]:

- based on code words (Fig. 2);
- based on syndromes (Fig. 3).

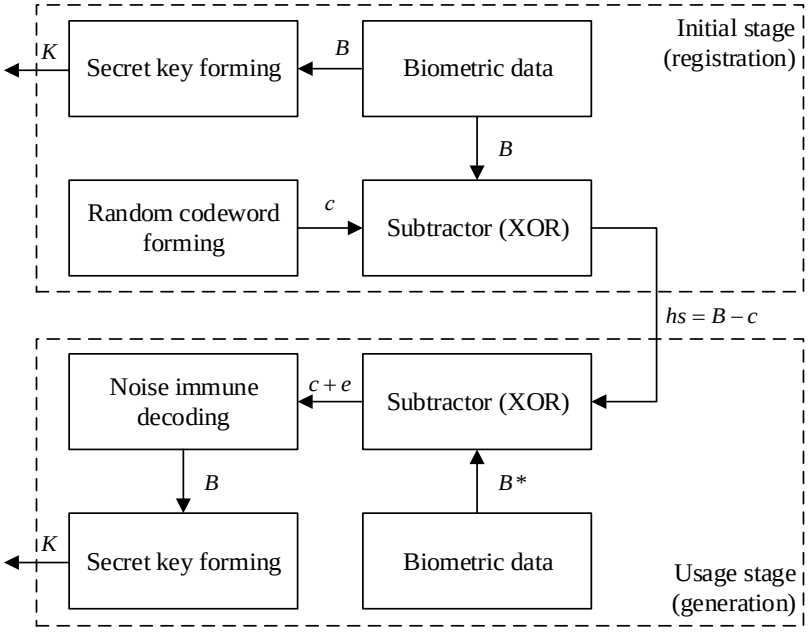


Fig. 2 – Scheme of fuzzy extractor based on codewords

Suppose that a noise immune block $(n,k,d=2t+1)$ code with error-correcting capability t is given. It is assumed that the presence of biometric data B allows the forming of a secret key K and some helper string hs (for this various techniques and techniques are used, for example, Secure Sketches [3,4]).

In the first construction (based on codewords, see Fig. 2) at the initial stage (*registration of the biometric key*) a random codeword c is formed. The open auxiliary line hs is formed by subtracting from the biometric data B the word c:

$$hs = B - c,$$

and by using this open line one can later restore the secret key K.

Indeed, during the usage phase the user provides biometric data B^*, from which the hint hs is subtracted. If $B^* \approx B$, then we have:

$$B^* - hs = B^* - (B - c) = c + e,$$

where $e = B^* - B$, and if Hamming weight of vector e does not exceed t, then decoding the vector $B^* - hs$ allows obtaining vectors c, e and therefore biometric data B:

$$B = c + hs.$$

Proper restoration of biometric data B allows you to generate a secret key K (*as in the registration phase*).

The second scheme (Fig. 3) operates syndrome sequences s that depend solely on the error vector e. To cite one example, for linear block codes given by a check matrix H, the following equalities hold for any codeword c [14,15]:

$$c \cdot H^T = 0,\ s = (c+e) \cdot H^T = e \cdot H^T.$$

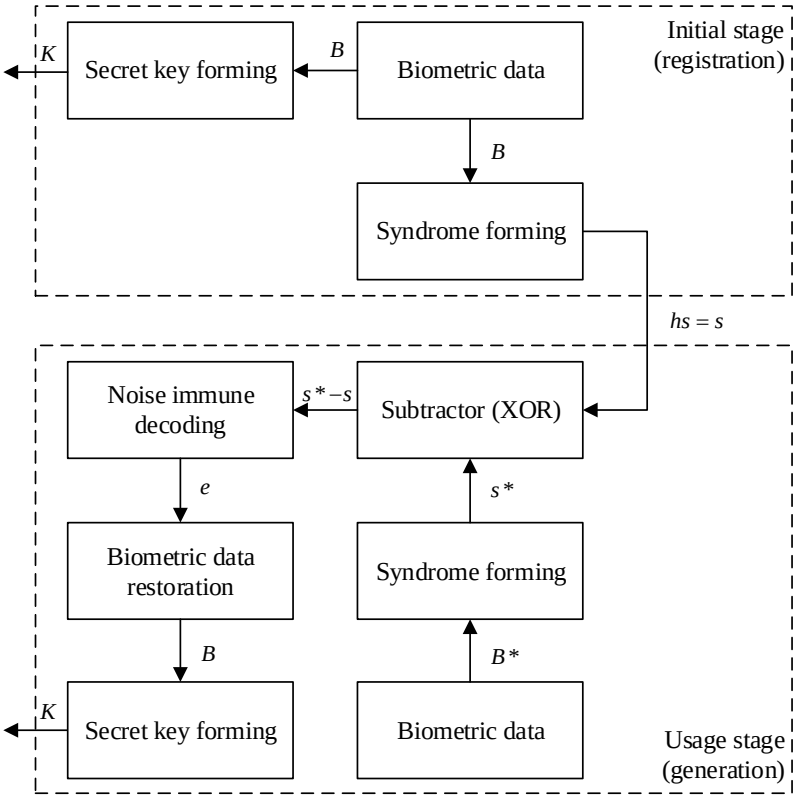


Fig. 3 – Scheme of fuzzy extractor based on syndromes

At the initial stage with use of B a syndrome sequence s is formed, which acts as open auxiliary data. At the stage of usage the user provides biometric data B^* for further syndrome sequence s^* computation. If $B^* \approx B$ then the hint $hs = s$ and the syndrome s^* allows you to restore B and generate the secret key K.

Indeed, if, for example, $s = B \cdot H^T$ and $s^* = B^* \cdot H^T$, then

$$s^* - s = e \cdot H^T,$$

where $e = B^* - B$, and if Hamming weight of vector e does not exceed t, then syndrome decoding of the vector $s^* - s$ allows to obtain e. Proper reconstruction of biometric data $B = B^* - e$ allows generation of secret key K (*as in the registration phase*).

It is obvious that the schemes in fig. 1 and fig. 2 for the binary case (addition and subtraction are realized by the XOR operation) practically coincide. The main difference is that in fig. 1 secret key is randomly generated, and then encoded with a noise immune code. In fig. 2 secret key is formed from biometric data B, which must be uniquely recovered in the event that the user provides the data $B^* \approx B$. However, in both schemes, a common approach is used, consisting in "blending" the biometric data B with the codeword c (randomly generated or encoded key K). This, in our opinion, can cause the main threat of using such biometric keys. If biometric data is transmitted, stored and / or processed in an open manner (*even mixed with codewords, syndromes, etc.*), then statistical attacks aimed at restoring code words c, biometric data B and keys K become possible.

In this paper we propose a new scheme for a fuzzy extractor, in which biometric data is not stored and transmitted in any form. This scheme uses the McEliece cryptosystem in the interpretation of Code-Based Electronic Digital Signature from [16].

3 Proposed scheme of Fuzzy Extractor

At the core of our proposal is the use of the McEliece cryptosystem [13].

McEliece code cryptosystem

McEliece cryptosystem was proposed in 1978 [13] and for 40 years of its existence it did not reveal any significant vulnerabilities. In the case of using Goppa codes [17] with sufficient length and

code distance, it is considered a reliable candidate for post-quantum application, i.e. it is supposed to be safe even if full-scale universal quantum computers are used to solve cryptographic analysis problems [18,19].

The public key in the McEliece scheme is the matrix

$$G_X = X \cdot G \cdot P \cdot D, \quad (1)$$

where G – generating matrix of an algebraic $(n,k,d=2t+1)$ code over $GF(q)$ (in the original paper [13] it was suggested to use the binary Goppa code [17]), X – a nonsingular matrix $k \times k$ with elements from $GF(q)$, P and D – permutational and diagonal $n \times n$ matrices (for binary codes only the matrix P is used).

Matrices X, P and D in (1) are a secret key that masks the algebraic block code used to match a random code (*general position code*), i.e. the public key G_X is available to the attacker as a randomly formed generating matrix of some linear code for which the fast decoding algorithm is unknown. On the contrary, an authorized user who knows the secret key (matrices X, P and D) can disable the action of the masking matrices and use the fast decoding algorithm for the algebraic code with the generating matrix G. A cryptogram is a vector of length n computed by rule

$$c_X^* = I \cdot G_X + e, \quad (2)$$

where vector

$$c_X = I \cdot G_X$$

is the codeword for the masked code, i.e. c_X belongs to $(n,k,d=2t+1)$ code with generating matrix G_X, I – k-bit information vector over $GF(q)$, vector e – secret error vector of weight t.

An attacker have to decode c_X^* using known to him matrix G_X. However, the decoding of a random code (with the corresponding parameters n,k,q and $d=2t+1$) is computationally unattainable. Since attacker doesn't know the matrices X, P and D, he can't recover the matrix G and use the decoding algorithm for polynomial complexity. For an authorized user (*who knows the secret key*) decoding is a polynomially solvable[2]. Indeed, an authorized user, having received a vector c_X^*, constructs a vector

$$\bar{c}^* = c_X^* \cdot D^{-1} \cdot P^{-1}. \quad (3)$$

Further, using an algorithm of polynomial complexity, he decodes a vector $\bar{c}^* = I' \cdot G + e'$, i.e. obtains I', then computes k-bit information vector

$$I = I' X^{-1}. \quad (4)$$

An additional secret parameter that can be used in the case of applying Goppa codes is the Goppa polynomial $G(x)$ [13].

New Code Based Fuzzy Extractor

The proposed scheme of fuzzy extractor allows generation of cryptographic keys even without using non-secret helper string. When using helper string, the proportion of corrected distortions of biometric images significantly increases. A simplified scheme of the proposed fuzzy extractor is shown in fig. 4.

At the initial stage, biometric data[3] B are interpreted as the codeword (2) of the masked code in the McEliece cryptosystem. In accordance with (3) its unmasking accomplished, the resulting vector $\bar{c}^*$ has been decoded. A vector I' is extracted from the decoded codeword, which is also unmasked in accordance with (4). The received information sequence I is interpreted as a secret biometric key K. In the simplest case $K = I$, although a more complex construct of generation K from I is possible, for example, unidirectional hashing: $K = h(I \| i)$, where: $x \| y$ – operation of concatenation of strings x and y; i –additional (service) data that is used to compute the secret key.

[2] For example, the Berlekamp-Massey decoding algorithm contains the number of multiplications of the order of t^2 [14, 15].

[3] It is assumed that at the registration stage the most reliable set of biometric characteristics is formed, represented as binary vectors

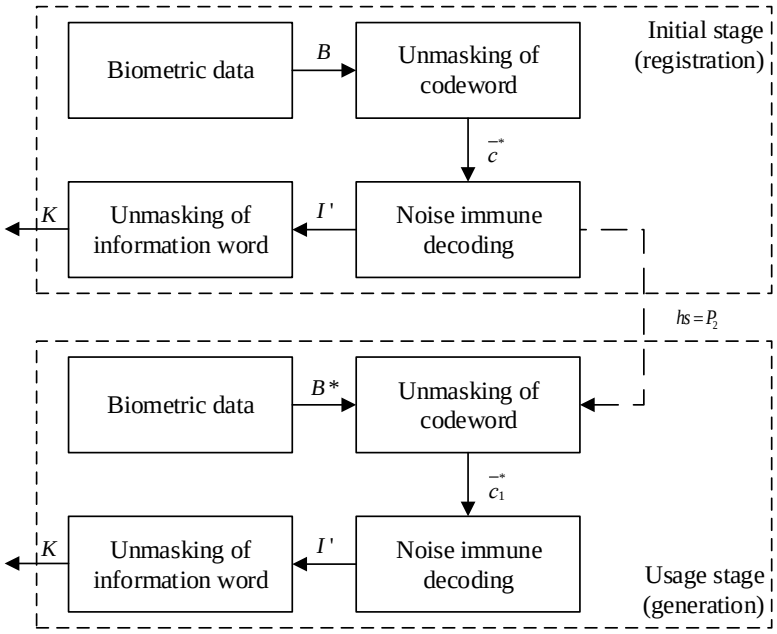


Fig. 4 – The proposed scheme of a fuzzy extractor
(a dashed line corresponds to the possible use of the helper string)

At the usage stage the user provides biometric data B^*, which, like at the registration stage, is interpreted as the codeword (2) of the masked code in the McEliece cryptosystem. In accordance with (3) it become unmasked, the resulting vector (denoted by $\vec{c}_1^*$) is decoded. If $B^* \approx B$ and, in our interpretation,

$$B = I \cdot G_X + e \text{ and } B^* = I \cdot G_X + e^*, \quad (5)$$

where e and e^* – two different vectors with Hamming weight less than t, then the decoding of vectors

$$\vec{c}^* = (I \cdot G_X + e) \cdot D^{-1} \cdot P^{-1} = I' \cdot G + e \cdot D^{-1} \cdot P^{-1}$$

and

$$\vec{c}_1^* = (I \cdot G_X + e^*) \cdot D^{-1} \cdot P^{-1} = I' \cdot G + e^* \cdot D^{-1} \cdot P^{-1}$$

will restore the same vector I'.

After unmasking the vector I' by rule (4), a secret key K is generated (as in the registration phase).

Our method is based on the assumption (5), that all belonging to the same user biometric characteristics have some general information (entropy), which can be notionally set by the vector I. This encoded information is distorted while processing of biometric images (*use of different biometric sensors, interference effects, erasures, etc.*). If we assume that biometric images are distorted by errors whose Hamming weight does not exceed the correcting ability t, then in all cases the secret key will be restored correctly. To reduce the effect of random errors in the registration phase, the most reliable set of biometric characteristics should be formed, for example, by multiple generations with averaging of obtained results.

The efficiency of the proposed fuzzy extractor, as well as other methods considered above, depends on the characteristics of the noise immune code it based on. In fact, False Rejection Rate (FRR) is determined by the probability of erroneous decoding (for case $B^* \approx B$). However, our assumption (5) looks more natural, the proposed extractor corrects various distortions of the same codeword containing biometric entropy. In the schemes [1] and [3,4] differences in the biometric patterns of the same user are corrected, i.e. the basic assumption underlying these constructions has the form where the Hamming weight of the vector must be smaller than t. If we take into account the possibility of multidirectional distortion of biometric images $B - B^*$, then our extractor intuitively appears more reliable.

It should be noted that in the scheme of fig. 4, the helper string is not obliged, i.e. the extractor can work "blindly". From each provided biometric image key data will be extracted and, if (5) is performed, the restored keys will be the same.

However, the additional use of helper string significantly reduces FRR.

Let's write the matrix G in the form of a "union" of two submatrices – a square $k \times k$ matrix G_1 and a rectangular $k \times (n-k)$ matrix G_2:

$$G = G_1 \| G_2 . \tag{6}$$

Then the $\bar{c} = I' \cdot G$ can be written in form:

$$\bar{c} = P_1 \| P_2 , \text{ where } P_1 = I' \cdot G_1 , \; P_2 = I' \cdot G_2 .$$

Using the last identities, we find P_2:

$$P_2 = P_1 \cdot G_1^{-1} \cdot G_2 , \tag{7}$$

where matrix G_1^{-1} is the inverse[4] to matrix G_1.

In fig. 4 the dashed line corresponds to the possible use of P_2 as a helper string (in the stage decoding of $\bar{c}_1^*$). This allows to reduce significantly the influence of errors and thus increase the probability of correct recovery of the vector I' and secret key K (i.e. reduce FRR). Indeed, if the errors (nonzero elements of the vector e) are distributed uniformly throughout the word $\bar{c}^* = I' \cdot G + e'$, then having an undistorted part P_2 of the codeword $I' \cdot G$, you can "ignore" all the errors that occur in the "second" part of the word. This is equivalent to increasing the correcting ability of the code according to the length of the vector P_2, as explained below.

Suppose the errors (non-zero elements of the vector e) occur randomly, equally and independently of each other. Denote by the symbol p the probability of distortion of single symbol of the codeword. Then the probability of distortion of m symbols of the code word of length n:

$$P(m) = C_n^m p^m (1-p)^{n-m} ,$$

where $C_n^m = \dfrac{n!}{m!(n-m)!}$ – binomial coefficient.

The probability of a decoding error (corresponds to FRR in our model without the use of helper string) when using the $(n, k, d = 2t+1)$ code can be written as:

$$FRR = 1 - \sum_{i=0}^{t} P(i) = 1 - \sum_{i=0}^{t} C_n^i p^i (1-p)^{n-i} . \tag{8}$$

When using helper string, errors need to be corrected only at the positions of the vector P_1 and the probability of decoding error (with similar reasoning) takes the form:

$$FRR^* = 1 - \sum_{i=0}^{t} C_k^i p^i (1-p)^{k-i} . \tag{9}$$

In fig. 5 showed the calculated dependences FRR for some (*n*, *k*, *d*) parameters of binary BCH codes: a) (127, 64, 21); b) (255, 115, 43); c) (512, 211, 83).

As follows from the dependencies above, when choosing the appropriate (*n*, *k*, *d)* parameters, the FRR can be very low. For example, when forming a 64-bit key using a binary (127, 64, 21) code and $p = 0,05$ the FRR value for an extractor without helper string does not exceed 10^{-1}. Using of helper string reduces FRR by 2 orders of magnitude. Increasing the length and correcting ability of the code results in a decrease of FRR. For example, for (512, 211, 83)-code even for $p = 0,15$ using helper string allows to form a 211 bit key with FAR no more than 10^{-1}.

[4] To invert the matrix G_1 it is necessary to correctly implement the representation (6): this is not a "union" of the first (any) k columns of the matrix G, but a pseudo-random choice of such k columns from form G which make a nonsingular square matrix G_1.

It should be noted that another important characteristic of biometric passwords – False Acceptance Rate (FAR), – characterizing the probability of incorrect secret key formation by an unauthorized user, increases with growing of code error-correct ability t. With a significant increase of t (due to increased redundancy P_2), the extractor will be able to extract the same key for any given biometric data, i. e. for $B^* \neq B$. For example, if you use a binary code with parameters (511, 112, 239) with a 399-bit hint $hs = P_2$, then even if all 112 bits of the vector P_1 are distorted, the extractor corrects them ($t = 119$) and unambiguously restores the vector I and secret key K. In other words, any user who provided an arbitrary set B^* can correctly recover the key K. From this point of view, when choosing (*n*, *k*, *d*) code parameters, a compromise solution should be chosen between the expected values of FRR and FAR.

Assuming that $k < \frac{n}{2}$ and all users have biometric data equidistant from each other, then FAR (with helper string) can be conditionally evaluated with the following expression:

$$FAR^* = \begin{cases} q^{-k+t}, & k > t; \\ 1, & k \le t, \end{cases} \tag{10}$$

where q – the power of the symbols alphabet over which the noise immune code is constructed (for binary code $q = 2$).

Indeed, in the proposed extractor, a vector I (*or a function of this vector*) of the length k of the code symbols is used as the secret key K. With equidistant code words (biometric images) and equiprobable choices, the probability of matching keys for different users is q^{-k}. The fuzzy extractor is based on noise immune decoding and, in the case of using the helper string, all t errors can be corrected on the block P_1 with the length k of code symbols, i.e. if $k > t$ then probability of coincidence of secret keys for different equiprobably selected biometric images will be equal q^{-k+t}. For $k \le t$ the correcting ability of the code allows to select completely the desired vector for any biometric set, i.e. "skipping the target" is a reliable event. If the helper string is not used, then for each k code characters there are $\frac{t}{n}$ errors which can be corrected in average and the FAR can be estimated as $q^{-k+\frac{t}{n}k}$.

Finally we should note that all the above reasoning, relations and computed values are given for "ideal" conditions, when sets of biometric characteristics are formed in form of binary vectors with random, equally probable (for $p < 0,5$) and independent errors. In real conditions the nature of errors can be significantly different. It is necessary to carry out further studies, including experimental ones, to provide practical recommendations on the direct use of the proposed fuzzy extractor.

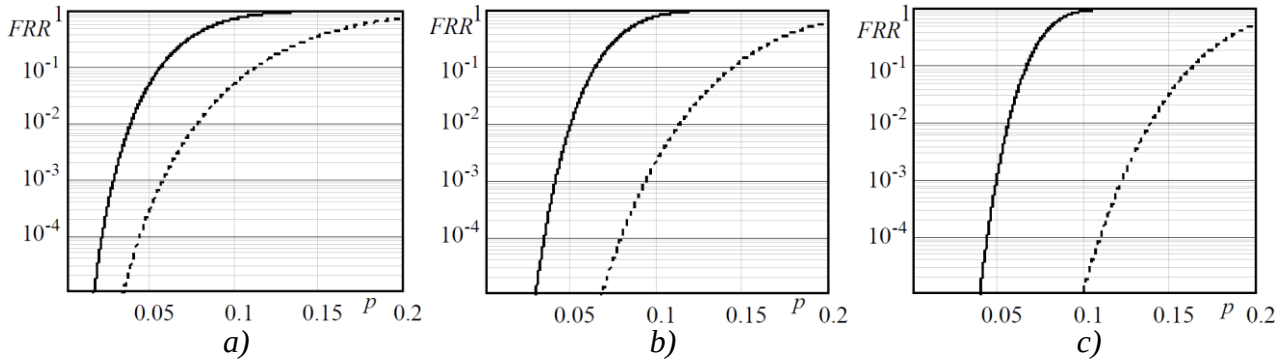


Fig. 5 – Computed dependencies FRR
(solid line – without helper string; dashed line – with helper string)

4 Conclusions

In this paper, a fuzzy extractor based on the McElice cryptosystem is proposed. Our proposal, on the one hand, uses the strengths of this code cryptosystem: cryptographic stability, based on the problem of syndrome decoding; resistance to quantum cryptanalysis methods; relatively high conversion speed (compared to other cryptosystems with a public key). On the other hand, proposed extractor by selecting the necessary (*n, k, d*) parameters of the noise immune code, allows to provide desired small FRRs (*admitting a number of assumptions about the nature of the errors*). The use of hints (*helper strings*) significantly reduces the FRR, but with the increase in the code correcting ability it can increase the FAR due to wrong "correction" of biometric features. The choice of a compromise solution concerning parameters of the code, taking into account the characteristics of the errors that arise, experimental studies of FRR and FAR are promising directions for further work.

References

[1] Hao F., Anderson R., Daugman J. Combining cryptography with biometrics effectively: Technical Report UCAM-CL-TR-640. Cambridge: University of Cambridge Computer Laboratory, 2005. 17 p.
[2] Juels A., Sudan M. A fuzzy vault scheme. Des. Codes Cryptography. 2006. Vol. 38, № 2. pp. 237–257.
[3] Fuzzy extractors: How to generate strong keys from biometrics and other noisy data /Dodis Y., Ostrovsky R., Reyzin L., Smith A. D. SIAM J. Comput. 2008. Vol. 38, № 1. pp. 97–139.
[4] Dodis Ye., Reyzin L., Smith A. Fuzzy Extractors. A Brief Survey of Results from 2004 to 2006. URL: http://www.cs.bu.edu/~reyzin/papers/fuzzysurvey.pdf
[5] Cryptographie key generation from PUF data using efficient fuzzy extractors/ Kang H., Hori Y., Katashita T., Hagiwara M. Iwamura K. 16th International Conference on Advanced Communication Technology. 2014, Pyeongchang. pp. 23–26.
[6] Fuzzy Extractors for Biometric Identification / Li N. and etc. 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). 2017, Atlanta, GA. pp. 667–677.
[7] Wen Y., Lao Y. Efficient fuzzy extractor implementations for PUF based authentication. 12th International Conference on Malicious and Unwanted Software (MALWARE). 2017, Fajardo. pp.119–125.
[8] Kaur T., Kaur M. Cryptographic key generation from multimodal template using fuzzy extractor. 2017 Tenth International Conference on Contemporary Computing (IC3). 2017, Noida. pp. 1–6.
[9] Gupta N. K. and Kaur M. A robust and secure multitrait based fuzzy extractor. *2017* 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT). 2017, Delhi. pp. 1–6.
[10] LWE-based lossless computational fuzzy extractor for the Internet of Things / Huth C. D. and etc. 2017 IEEE International Symposium on Hardware Oriented Security and Trust (HOST). 2017, McLean, VA. P. 154.
[11] Securing Systems With Indispensable Entropy: LWE-Based Lossless Computational Fuzzy Extractor for the Internet of Things / Huth C. and etc. IEEE Access. 2017. Vol. 5. pp. 11909–11926.
[12] Eliminating Leakage in Reverse Fuzzy Extractors / Schaller A., Stanko T., Škorić B. and Katzenbeisser S. IEEE Transactions on Information Forensics and Security. 2018. Vol. 13, № 4. pp. 954–964.
[13] McEliece R. J. A public-key cryptosystem based on algebraic coding theory. DSN Progress Report 42-44, Jet Propulsion Lab. 1978, Pasadena, CA. pp. 114–116.
[14] Clark G.C., Cain J.B. Error-Correction Coding for Digital Communications. Springer, 1981. 432 p.
[15] Blahut R. E. Theory and Practice of Error Control Codes. Massachusetts: Addison Wesley Publishing Company Inc., 1983. 500 p.
[16] Code-based electronic digital signature/ Kuznetsov A., Pushkar'ov A., Kiyan N., Kuznetsova T. 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT). 2018. Kyiv. pp. 331–336.
[17] Goppa V.D. A New Class of Linear Correcting Codes. Problems Inform. Transmission. 1970. Vol. 6, № 3. pp. 207–212.
[18] Bernstein D., Buchmann J., Dahmen E. Post-Quantum Cryptography. Berlin-Heidleberg: Springer-Verlag, 2009. 245 p.

Рецензент: Сергій Кавун, доктор економічних наук, к.т.н., проф., Харківський технологічний університет "ШАГ", вул. Малом'ясницька, 9/11, м. Харків, 61010, Україна.
E-mail: kavserg@gmail.com

Надійшло: Серпень 2018.

Автори:
Олександр Кузнецов, д.т.н., проф., академік Академії наук прикладної радіоелектроніки, Харківський національний університет імені В.Н. Каразіна, майдан Свободи 6, м. Харків, 61022, Україна. E-mail: kuznetsov@karazin.ua

Анастасія Киян, студентка, Харківський національний університет імені В.Н. Каразіна, майдан Свободи 6, м. Харків, 61022, Україна. E-mail: nastyak931@gmail.com

Роман Сергієнко, Національна академія сухопутних військ імені гетьмана Петра Сагайдачного, вул. Героїв Майдану 32, м. Львів, 79012, Україна.
E-mail: romanserg69@gmail.com

Анна Уварова, провідний інженер, Конструкторське бюро «Південне» ім. М. К. Янгеля», вул. Криворізька 3, м. Дніпро, 49008, Україна. E-mail: annet.uvarova@gmail.com

Дмитро Прокопович-Ткаченко, к.т.н., завідувач кафедрою кібербезпеки, Університет митниці та фінансів, вул. Володимира Вернадського 2/4, м. Дніпро, 49000, Україна. E-mail: omega2@email.dp.ua

Екстрактор біометричних ключів на кодових криптосистемах.

Анотація. У даній роботі розглядаються методи формування криптографічних ключів з біометричних образів із використанням нечітких екстракторів. Пропонується нова схема нечіткого екстрактора, в основі якої лежить кодова криптосистема Мак-Еліса. Показано, що нова конструкція нечіткого екстрактора дозволяє формувати криптографічні паролі з біометричних образів навіть без використання несекретної підказки (допоміжного рядка). При використанні допоміжної рядка значно зростає частка коректованих спотворень біометричних образів. Крім того, пропонована конструкція відноситься до класу пост-квантових методів захисту інформації, тобто очікується її безпечне використання навіть в умовах застосування універсальних квантових комп'ютерів для вирішення завдань криптоаналізу.

Ключові слова: криптосистема на основі коду; нечіткий екстрактор; біометрична криптографія; криптографічні ключі.

Рецензент: Сергей Кавун, доктор экономических наук, к.т.н., проф., Харьковский технологический университет «ШАГ», ул. Маломясницкая, 9/11, г. Харьков, 61010, Украина.
E-mail: kavserg@gmail.com

Поступила: Август 2018.

Авторы:

Александр Кузнецов, д.т.н., проф., Академик Академии наук прикладной радиоэлектроники, Харьковский национальный университет им. В. Н. Каразина, пл. Свободы 6, г. Харьков, 61022, Украина. E-mail: kuznetsov@karazin.ua

Анастасия Киян, студентка, Харьковский национальный университет имени В.Н. Каразина, пл. Свободы 6, г. Харьков, 61022, Украина. E-mail: nastyak931@gmail.com

Роман Сергиенко, Национальная академия сухопутных войск имени гетмана Петра Сагайдачного, ул. Героев Майдана, 32, г. Львов, 79012, Украина. E-mail: romanserg69@gmail.com

Анна Уварова, ведущий инженер, Конструкторское бюро «Южное» им. М. К. Янгеля», г. Днепр, ул. Криворожская 3, 49008, Украина. E-mail: annet.uvarova@gmail.com

Дмитрий Прокопович-Ткаченко, к.т.н., заведующий кафедрой кибербезопасности, Университет таможни и финансов, ул. Владимира Вернадского, 2/4, г. Днепр, 49000, Украина. E-mail: omega2@email.dp.ua

Экстрактор биометрических ключей на кодовых криптосистемах.

Аннотация. В данной работе рассматриваются методы формирования криптографических ключей из биометрических образов с использованием нечетких экстракторов. Предлагается новая схема нечеткого экстрактора, в основе которой лежит кодовая криптосистема Мак-Элиса. Показано, что новая конструкция нечеткого экстрактора позволяет формировать криптографические пароли из биометрических образов даже без использования несекретной подсказки (вспомогательной строки). При использовании вспомогательной строки значительно возрастает доля корректируемых искажений биометрических образов. Кроме того, предлагаемая конструкция относится к классу пост-квантовых методов защиты информации, т.е. ожидается ее безопасное использование даже в условиях применения универсальных квантовых компьютеров для решения задач криптоанализа.

Ключевые слова: криптосистема на основе кода; нечеткий экстрактор; биометрическая криптография; криптографические ключи.

UDC 004.056.55

STATISTICAL PROPERTIES OF MODERN STREAM CIPHERS

Oleksii Nariezhnii, Egor Eremin, Vladislav Frolenko, Kyrylo Chernov, Tetiana Kuznetsova, Yevhen Demenko

V. N. Karazin Kharkiv National University, 6 Svobody Sq., Kharkiv, 61022, Ukraine
o.nariezhnii@karazin.ua, suvenick2@gmail.com, jadson27101@gmail.com, kirillfilippsky@gmail.com, kuznetsova.tatiana17@gmail.com, demenjay@gmail.com

Reviewer: Ivan Gorbenko, Doctor of Sciences (Engineering), Full Professor, Academician of the Academy of Applied Radioelectronics Sciences, V. N. Karazin Kharkiv National University, 4 Svobody Sq., Kharkiv, 61022, Ukraine
gorbenkoi@iit.kharkov.ua

Received on September 2018

Abstract. *In recent years, numerous studies of stream symmetric ciphers in Ukraine are continuing, the main purpose of which is to argue the principles of creating a new cryptographic algorithm, which can be based on the national standard. One of the essential aspects in choosing from many alternatives is the statistical properties of the output pseudorandom sequence (key stream). In this paper, the results of comparative studies of statistical properties of output sequences, which are formed by various stream ciphers, in particular, by world-known algorithms Enocoro, Decim, Grain, HC, MUGI, Mickey, Rabbit, RC-4, Salsa20, SNOW2.0, Sosemanuk, Trivium and the Ukrainian cryptographic algorithm Strumok, that was developed in recent years, are presented. For comparative studies, the NIST STS method was used, according to which experimental studies are performed in 15 statistical tests, the purpose of which is to determine the randomness of the output binary sequences. Each of the tests is aimed at studying certain vulnerabilities of the generator, that is, points to the potential usage of different methods of cryptographic analysis. Although each of the considered streaming encryption algorithms has been studied, we have carried out a statistical test of the generated pseudorandom sequences under equal conditions and with identical initial parameters, that is, our results allow us to perform a comparative analysis of ciphers and to justify the best of statistical properties. The estimates presented in the article, as expected, confirmed the high statistical security indexes of modern ciphers. In addition, according to the results of experimental research, it was found that the new Ukrainian development - the stream cipher Strumok does not yield to the best world algorithms in the statistical properties of the initial sequences*.

Keywords: *symmetric cryptography; stream cryptographic algorithms; gamma; cryptanalysis; statistical tests.*

1 Introduction

Nowadays, there is a large number of areas in technology where it is necessary to use cryptography to protect information. These areas include Internet of things (IoT), smart sensors and controllers, medical devices, RFID tags. That is why in the modern world different requirements for crypto algorithms are formed, however there is one common requirement – it is stability.

The stability of the stream cryptosystem depends very much on the quality of the generated gamma, because if the gamma is predictable, the space of the possible keys will be smaller and it will be possible to go through all values. Determining the randomness of the generated sequences is one of the main tasks.

Generating random numbers means getting a sequence of binary characters 0 and 1. Pseudorandom sequence generators are a function that is initialized by the seed, and then generate a sequence 0 and 1 at the output. Knowing this seed, we can predict the entire sequence. A good PRNG generator is one for which it is impossible to predict the following values, with having all the history of the previous values without having the seed [1].

This article is dedicated to the study of statistical properties of pseudorandom sequences, which are formed by different streaming cryptographic algorithms.

For conducting experimental studies, world-known stream symmetric ciphers that were standardized at the international and/or national level or presented in various research projects, for example, at the international eSTREAM competition, were involved. In particular, we reviewed the specification and developed the software implementation of stream ciphers Enocoro, Decim, Grain, HC, MUGI, Mickey, Rabbit, RC-4, Salsa20, SNOW2.0, Sosemanuk, Trivium [2-7]. The Ukrainian stream cipher Strumok, which was developed and presented in [8] and subsequently improved several times and researched [9-13], was involved to studies in addition. In particular, in our last paper

[13] the same streaming cryptographic algorithms were considered, but we carried out a comparative analysis of the computational complexity of stream ciphers and their performance on various computing systems.

The test of the National Institute of Standards and Technology (NIST) of the USA «A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications» (STS) [1] was used in this work to conduct comparative experimental studies of the statistical properties of output pseudorandom sequences. This method has been developed and tested in the study of candidates for the national standard of block symmetric ciphers in the United States (the winner and published as standard [14], as is known, was the algorithm Rijndael).

The NIST STS method consists of 15 independent statistical tests, although, depending on individual parameters and settings, 188 statistical tests are usually performed. The basis of all these tests is the concept of zero hypothesis. Zero hypothesis is the assumption that there is no relationship between the two facts. There is also an alternative hypothesis that refutes the null hypothesis: that is, there is an interconnection between facts. If we proceed to the terms of random numbers, then for the null hypothesis assumes that the sequence is truly random (whose values appear equip probable and independently of each other). Consequently, if the null hypothesis is correct, then our generator produces sufficiently "good" random numbers [1]. For each test, we need to select the appropriate random statistics and use it to determine whether or not to accept the null hypothesis. With the assumption of randomness, such statistics have a distribution of possible values. The theoretical proof of the distribution of this statistics for a zero hypothesis is determined by mathematical methods. During the test, the value of the statistic for the data (the test sequence) is calculated. This statistic value of the test is compared with the critical value. If the value of the statistical test exceeds the critical value, the null hypothesis of randomness is rejected. Otherwise, the null hypothesis (the hypothesis of randomness) does not deviate (that is, the hypothesis is accepted) [1].

Of course, chosen cryptographic transformations Enocoro, Decim, Grain, HC, Mugi, Mickey, Rabbit, RC-4, Salsa20, SNOW2.0, Sosemanuk and Trivium have been repeatedly tested, including by developers [2-7]. But all these studies were performed on different platforms with different settings and output parameters. That is, the statistical properties of the initial sequences of each algorithm (*with their own settings*) were studied without conducting a comparative analysis of the statistical safety indicators and the justification of the best cipher according to the statistical properties.

The purpose of our work is to independently test the properties of cryptographic algorithms from [2-7] with universal and identical conditions in order to obtain objective and independent results. Special attention was paid to the streaming encryption algorithm Strumok, which was developed and presented by Ukrainian scientists in [8-13]. To date, numerous studies have been conducted in Ukraine to substantiate the principles of creating a new streaming cryptographic algorithm, which may be the base of the national standard. Therefore, experimental studies of the statistical properties of the Strumok algorithm, their comparative analysis with the value of statistical security of world-known ciphers from [2-7] is undoubtedly relevant and important scientific task.

2 Researched Symmetric Ciphers and Testing Conditions

During conducting of experimental researches modern stream symmetric ciphers were considered: Enocoro [2,3], Decim [2,6], Grain [7], HC [7], MUGI [2], Mickey [7], Rabbit [2,7], RC-4 [7], Salsa20 [7], SNOW2.0 [2], Sosemanuk [7], Trivium [2], Strumok [8,11], also block encryption algorithm AES [14,15], which can be used in streaming encryption modes.

The list of studied algorithms is given in Table 1, which provides brief information about ciphers and affiliation with relevant standards or research projects.

For testing were developed software implementation of all the studied stream ciphers in Java language. For each crypto algorithm, sequences of size of one million bits and ten thousand bits were generated with the help of a randomly generated key. These sequences have been tested on a single platform by tests on randomness.

These tests focus on a types of non-randomness that can exist in the sequence.

Table 1 – The Chosen Crypto Algorithms For Comparison

The name of the cipher	Source of specification	State size, bit	Key size, bit	Size IV, bit
AES	FIPS-197, CRYPTREC, ISO/IEC 18033-4	128	128, 256	256
Enocoro	ISO/IEC 29192-3	272	80, 128	64
DECIMv2	ISO/IEC 18033-4, eSTREAM	288	128	128
GRAIN	eSTREAM	128	128	96
HC	eSTREAM	128, 256	128, 256	128, 256
MUGI	ISO/IEC 18033-4	128	128	128
MICKEY	eSTREAM	160	128	128
Rabbit	ISO/IEC 18033-4, eSTREAM	513	128	64
RC4	Mailing list Cypherpunks	256	256	–
SALSA-20	eSTREAM	512	128	64
SNOW2.0	ISO/IEC 18033-4	512	128, 256	128, 56
SOSEMANUK	eSTREAM	512	128	128
TRIVIUM	eSTREAM, ISO/IEC 29192-3	288	80	80
Strumok-256	[8, 11]	1024	256	256
Strumok-512		1024	512	512

These tests are included: The Frequency (Monobit) Test, Frequency Test within a Block, The Runs Test, Tests for the Longest-Run-of-Ones in a Block, The Binary Matrix Rank Test, The Discrete Fourier Transform (Spectral) Test, The Non-overlapping Template Matching Test, The Overlapping Template Matching Test, Maurer's "Universal Statistical" Test, The Linear Complexity Test, The Serial Test, The Approximate Entropy Test, The Cumulative Sums (Cusums) Test, The Random Excursions Test, and The Random Excursions Variant Test [1].

For all tests, the rule of acceptance the randomness is as follows: if the computed P-value is < 0.01, then conclude that the sequence is non-random. Otherwise, conclude that the sequence is random [1].

Randomness tests were implemented in Python. For each test, a quantitative probability characteristic was obtained, these values were studied and analyzed. The experimental results obtained from each test were compared with the value of the criterion of significance (which is equal to 0.01, that is, it is a probability to reject the correct zero hypothesis by mistake).

3 Results of Statistical Tests

The results of NIST testing are shown in Tables 2-16.

Results of The Frequency (Monobit) Test and Results of Frequency Test within a Block (Table 2, 3) can be interpreted as follows: the closer the value of P-value to 1 is more evenly distributed "0" and "1". A small value of P-value means a large deviation from the equable distribution of zeros and ones in at least one of the sequence blocks. That is, the smaller the value, the more unevenly distributed values.

Table 2 – Results of The Frequency (Monobit) Test

The name of the cipher	P-value	The name of the cipher	P-value
AES-128	0.52217	Rabbit	0.92034
AES-256	0.98404	RC4	0.01046
Enocoro	0.20054	SALSA-20	0.33705
DECIM v2	0.52217	SNOW2.0-128	0.81033

Continuation of Table 2

GRAIN	0.20059	SNOW2.0-256	0.14986
HC -128	0.14429	SOSEMANUK	0.07840
HC -256	0.48392	TRIVIUM	0.50925
MUGI	0.92034	Strumok-256	1.0
MICKEY	0.49650	Strumok-512	0.14135

Table 3 – Results of the Frequency Test within a Block

The name of the cipher	P-value	The name of the cipher	P-value
AES-128	0.35359	Rabbit	0.46658
AES-256	0.25300	RC4	0.50833
Enocoro	0.39109	SALSA-20	0.71156
DECIM v2	0.33150	SNOW2.0-128	0.86422
GRAIN	0.64794	SNOW2.0-256	0.06933
HC -128	0.96846	SOSEMANUK	0.72974
HC -256	0.92374	TRIVIUM	0.51628
MUGI	0.40069	Strumok-256	0.75319
MICKEY	0.64356	Strumok-512	0.36249

Table 4 – Result of the Runs Test

The name of the cipher	P-value	The name of the cipher	P-value
AES-128	0.12254	Rabbit	0.23795
AES-256	0.54850	RC4	0.70787
Enocoro	0.06926	SALSA-20	0.97669
DECIM v2	0.39203	SNOW2.0-128	0.28887
GRAIN	0.51188	SNOW2.0-256	0.96867
HC -128	0.65891	SOSEMANUK	0.34245
HC -256	0.07756	TRIVIUM	0.17519
MUGI	0.77940	Strumok-256	0.71884
MICKEY	0.23191	Strumok-512	0.56320

Table 5 – Results of Tests for the Longest-Run-of-Ones in a Block

The name of the cipher	P-value	The name of the cipher	P-value
AES-128	0.56531	Rabbit	0.30447
AES-256	0.31757	RC4	0.60474
Enocoro	0.32566	SALSA-20	0.54449
DECIM v2	0.45269	SNOW2.0-128	0.21839
GRAIN	0.46634	SNOW2.0-256	0.59095
HC -128	0.22625	SOSEMANUK	0.17928
HC -256	0.50297	TRIVIUM	0.12140
MUGI	0.18882	Strumok-256	0.61075
MICKEY	0.16730	Strumok-512	0.24879

Table 6 – Results of the Binary Matrix Rank Test

The name of the cipher	P-value	The name of the cipher	P-value
AES-128	0.62395	Rabbit	0.53858
AES-256	0.34227	RC4	0.37364
Enocoro	0.42643	SALSA-20	0.62596
DECIM v2	0.53868	SNOW2.0-128	0.42577

Continuation of Table 6

GRAIN	0.25851	SNOW2.0-256	0.56234
HC -128	0.35643	SOSEMANUK	0.23127
HC -256	0.63508	TRIVIUM	0.32610
MUGI	0.25313	Strumok-256	0.77838
MICKEY	0.43287	Strumok-512	0.43683

Table 7 – Results of the Discrete Fourier Transform (Spectral) Test

The name of the cipher	P-value	The name of the cipher	P-value
AES-128	0.58190	Rabbit	0.78308
AES-256	0.40886	RC4	0.58190
Enocoro	0.46286	SALSA-20	0.19888
DECIM v2	0.04350	SNOW2.0-128	0.40886
GRAIN	0.77958	SNOW2.0-256	0.82688
HC -128	0.58190	SOSEMANUK	0.71357
HC -256	0.14203	TRIVIUM	0.92688
MUGI	1.0	Strumok-256	0.47081
MICKEY	0.40886	Strumok-512	0.58190

Table 8 – Results of the Non-overlapping Template Matching Test

The name of the cipher	P-value	The name of the cipher	P-value
AES-128	0.88179	Rabbit	0.53417
AES-256	0.58654	RC4	0.74029
Enocoro	0.32005	SALSA-20	0.51304
DECIM v2	0.51093	SNOW2.0-128	0.29576
GRAIN	0.03834	SNOW2.0-256	0.98602
HC -128	0.72360	SOSEMANUK	0.71631
HC -256	0.66453	TRIVIUM	0.61436
MUGI	0.57822	Strumok-256	0.88153
MICKEY	0.70717	Strumok-512	0.70137

The smaller the value of P-value for the Runs Test (Table 4) the smaller the number of changes of the value in the sequence. Fluctuations should be approximately equal to the expected fluctuations in completely random sequences.

The greater the value of P-value for the greater the value of P-value (Table 5) the more evenly distributed cluster ones in the sequence. The small values of the P-value for the Binary Matrix Rank Test (Table 6) indicate that there is a deviation of the rank distribution from what corresponds to the random sequence. With a low value of the P-value for The Discrete Fourier Transform (Spectral) Test (Table 7), we can conclude that in the sequence there are too many peaks that exceed the threshold in the sequence if it was random.

If the P-value is too small for The Overlapping Template Matching Test (Table 9), then in sequence there is an excess of defined value patterns. According to the results of this test, we can also say if there are too many single defined patterns in the sequence. This will be visible if the P-value is too small.

Table 9 – Results of the Overlapping Template Matching Test

The name of the cipher	P-value	The name of the cipher	P-value
AES-128	0.63467	Rabbit	0.46289
AES-256	0.42685	RC4	0.64860
Enocoro	0.35152	SALSA-20	0.49497

Continuation of Table 9

DECIM v2	0.52759	SNOW2.0-128	0.64848
GRAIN	0.24286	SNOW2.0-256	0.52567
HC -128	0.58395	SOSEMANUK	0.64954
HC -256	0.75296	TRIVIUM	0.38551
MUGI	0.48632	Strumok-256	0.83526
MICKEY	0.63858	Strumok-512	0.53734

Table 10 – Results of the Maurer's "Universal Statistical" Test

The name of the cipher	P-value	The name of the cipher	P-value
AES-128	0.72412	Rabbit	0.48576
AES-256	0.52788	RC4	0.40352
Enocoro	0.32640	SALSA-20	0.73056
DECIM v2	0.42902	SNOW2.0-128	0.58112
GRAIN	0.57329	SNOW2.0-256	0.68674
HC -128	0.66285	SOSEMANUK	0.59053
HC -256	0.42758	TRIVIUM	0.48921
MUGI	0.74892	Strumok-256	0.64992
MICKEY	0.67342	Strumok-512	0.57924

Table 11 – Results of the Linear Complexity Test

The name of the cipher	P-value	The name of the cipher	P-value
AES-128	0.68768	Rabbit	0.36868
AES-256	0.65642	RC4	0.56226
Enocoro	0.86532	SALSA-20	0.35788
DECIM v2	0.46743	SNOW2.0-128	0.38573
GRAIN	0.44180	SNOW2.0-256	0.57693
HC -128	0.45474	SOSEMANUK	0.45798
HC -256	0.64957	TRIVIUM	0.55617
MUGI	0.95746	Strumok-256	0.73563
MICKEY	0.57499	Strumok-512	0.86376

Table 12 – Results of the Serial Test

The name of the cipher	P-value	The name of the cipher	P-value
AES-128	0.68433	Rabbit	0.34335
AES-256	0.54761	RC4	0.63467
Enocoro	0.54739	SALSA-20	0.52347
DECIM v2	0.44883	SNOW2.0-128	0.36832
GRAIN	0.35638	SNOW2.0-256	0.78322
HC -128	0.57221	SOSEMANUK	0.34567
HC -256	0.23488	TRIVIUM	0.67347
MUGI	0.34572	Strumok-256	0.72752
MICKEY	0.78763	Strumok-512	0.62366

Analyzing the value of P-value by the results of the Maurer's "Universal Statistical" Test (Table 10), one can say whether the sequence is too compressed. The greater the value of P-value, the less compresable is the sequence. As a result of the Linear Complexity Test (Table 11), it is possible to determine whether the linear shift register exists in the sequence generator. The higher the value of P-value – the larger the size of the shift generator is required to reconstruct the sequence.

For random sequences, this size is about half its length. For a small value of P-value for tests from Table 12, the heterogeneity of the sequence sub-blocks is implicitly indicated.

Table 13 – Results of the Approximate Entropy Test

The name of the cipher	P-value	The name of the cipher	P-value
AES-128	0.27357	Rabbit	0.34791
AES-256	0.39911	RC4	0.34671
Enocoro	0.12951	SALSA-20	0.86884
DECIM v2	0.27339	SNOW2.0-128	0.10917
GRAIN	0.19162	SNOW2.0-256	0.79903
HC -128	0.33410	SOSEMANUK	0.61877
HC -256	0.07352	TRIVIUM	0.52246
MUGI	0.36244	Strumok-256	0.41764
MICKEY	0.09856	Strumok-512	0.31153

Table 14 – Results of the Cumulative Sums (Cusums) Test

The name of the cipher	P-value	The name of the cipher	P-value
AES-128	0.42853	Rabbit	0.68396
AES-256	0.64328	RC4	0.53665
Enocoro	0.37563	SALSA-20	0.62532
DECIM v2	0.42514	SNOW2.0-128	0.47863
GRAIN	0.58636	SNOW2.0-256	0.54796
HC -128	0.68527	SOSEMANUK	0.44387
HC -256	0.47623	TRIVIUM	0.58935
MUGI	0.57836	Strumok-256	0.79836
MICKEY	0.36568	Strumok-512	0.59623

Table 15 – Results of the Random Excursions Test

The name of the cipher	P-value	The name of the cipher	P-value
AES-128	0.67548	Rabbit	0.73477
AES-256	0.77585	RC4	0.34773
Enocoro	0.34648	SALSA-20	0.75442
DECIM v2	0.46672	SNOW2.0-128	0.83739
GRAIN	0.57847	SNOW2.0-256	0.63478
HC -128	0.34678	SOSEMANUK	0.42488
HC -256	0.87343	TRIVIUM	0.45547
MUGI	0.49898	Strumok-256	0.84743
MICKEY	0.73987	Strumok-512	0.62398

Table 16 – Results of the Random Excursions Variant Test

The name of the cipher	P-value	The name of the cipher	P-value
AES-128	0.59235	Rabbit	0.58363
AES-256	0.40247	RC4	0.35872
Enocoro	0.13462	SALSA-20	0.54290
DECIM v2	0.38366	SNOW2.0-128	0.68125
GRAIN	0.48532	SNOW2.0-256	0.47193
HC -128	0.59291	SOSEMANUK	0.74821
HC -256	0.48363	Strumok-256	0.68223
MUGI	0.59254	Strumok-512	0.51944
MICKEY	0.39660	TRIVIUM	0.39576

The greater the value of P-value by the result of the test for approximate entropy (Table 13), the greater the degree of uncertainty of the values in the sequence. The smaller the value of P-value in Table 14, the greater the excessive uniformity of the distribution of values over the sequence. If the value of P-value from table 15 is too small, then it means that there is a deviation of the distribution of the selected template in all sub-blocks of the sequence. Finally, for the Random Excursions Variant Test (Table 16): if the value of P-value is too small then it means that there is a deviation of the distribution of the selected template in all subsequences. Thus, as shown in tables 2-16, all researched cryptographic algorithms have good statistical properties. In particular, according to the criterion "P-value < 0,01" it is concluded that all the formed sequences are random, and the corresponding generators can be used for cryptographic purposes.

The results of experimental statistical studies generally coincide with published earlier data and conclusions for the studied stream ciphers Enocoro, Decim, Grain, HC, MUGI, Mickey, Rabbit, RC-4, Salsa 20, SNOW 2.0, Sosemanuk, Trivium [2-7] and block algorithm AES [14,15]. However, according to the results of the comparative analysis values of statistical safety, ciphers SNOW 2.0 and Salsa 20 should be marked as the best. In addition, experimental studies of the output sequences of the Ukrainian algorithm Strumok [8,11] showed that according to statistical properties it is not inferior to world-known cryptographic algorithms.

4 Conclusions

In this work, an independent statistical testing of the most well-known stream symmetric ciphers, standardized at the national and / or international level, or presented as winners of international search projects, has been conducted. Our researches were conducted on equal terms and with identical initial parameters for all the ciphers under study. To do this, we used our own software implementation of cryptographic algorithms and the NIST STS statistical test suite.

During the execution of 15 statistical tests of NIST STS, the randomness of the binary sequences of modern stream symmetric ciphers was determined. Presented estimates, as expected, confirmed the high levels of their statistical security. According to the results of this work, it can be said that modern cryptographic algorithms fully meet the requirements for randomness and, as a consequence, reliable. However, it's important to note that there are algorithms that have shown better results throughout all tests, that is: SNOW 2.0 and Salsa 20.

It should be noted that the algorithm Strumok gives quite acceptable results in comparison with world analogues. This indicates that the sequence generated by the Strumok stream algorithm is close to random, that is, it is safe to use for cryptographic purposes.

The prospects for further research include a detailed analysis of the ciphers for the possibility of performing cryptanalytic attacks. It is advisable to investigate the ability of the considered ciphers, including the Strumok algorithm, to function reliably under post-quantum conditions, the main problems and prospects of which are studied in various works, for example, in [16-24].

References

[1] A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. NIST Special Publication 800-22, 2010. URL: https://dl.acm.org/citation.cfm?id=2206233

[2] Information technology. Security techniques. Encryption algorithms. Part 4: Stream ciphers. ISO/IEC 18033-4, 2011. URL: https://www.iso.org/standard/54532.html

[3] Information technology. Security techniques. Lightweight cryptography. Part 3: Stream ciphers. ISO/IEC 29192-3, 2012. URL: https://www.iso.org/standard/56426.html

[4] Pseudorandom Number Generator Enocoro.URL: http://www.cryptrec.go.jp

[5] Decim – A new Stream Cipher for Hardware applications. ECRYPT Stream Cipher Project Report 2005/004. URL: http://www.ecrypt.eu.org/

[6] Hongjun W., Preneel B. Cryptanalysis of Stream Cipher Decim. URL: http://www.ecrypt.eu.org/stream/

[7] The eSTREAM Project. URL: http://www.ecrypt.eu.org/

[8] Kuznetsov O., Lutsenko M., Ivanenko D. Strumok stream cipher: Specification and basic properties. 2016 Third International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T). Kharkiv, 2016, p. 59-62.

[9] Kuznetsov A., Kolovanova Y., Kuznetsova T. Periodic characteristics of output feedback encryption mode. 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T). Kharkov, 2017. pp. 193-198.

[10] The research of modern stream ciphers / Gorbenko I., Kuznetsov A., Lutsenko M., Ivanenko D. 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T). Kharkov, 2017. pp. 207-210.
[11] Strumok keystream generator / Gorbenko I. and etc. 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT). 2018. pp. 294-299.
[12] Analysis of block symmetric algorithms from international standard of lightweight cryptography ISO/IEC 29192-2/ Kuznetsov A., Gorbenko Y., Andrushkevych A. , Belozersev I. 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T). Kharkov, 2017. pp. 203-206.
[13] Research of cross-platform stream symmetric ciphers implementation/ Kuznetsov A., Frolenko V., Eremin E., Zavgorodnia O. 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT). Kyiv, 2018. pp. 300-305.
[14] FIPS–197: Advanced Encryption Standard (AES). National Institute of Standards and Technology, 2001. URL: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
[15] ISO/IEC 18033-3:2010. Information technology – Security techniques – Encryption algorithms – Part 3: Block ciphers. URL: https://www.iso.org/standard/54531.html
[16] Deutsch D., Jozsa R. Rapid solutions of problems by quantum computation. Proceedings of The Royal Society of London. A: Mathematical, Physical and Engineering Sciences.1992. Vol. 439, №1907. pp. 553-558.
[17] Bernstein D., Buchmann J. , Dahmen E. Post-Quantum Cryptography. Berlin-Heidleberg: Springer-Verlag, 2009. 245 p.
[18] Post-Quantum Crystography: A combination of Post-Quantum Cryptography and Steganography/ Gabriel A. J., Alese B. K., Adetunmbi A. O., Adewale O. S. 8th International Conference for Internet Technology and Secured Transactions (ICITST-2013). London, 2013. pp. 449-452.
[19] Code-based public-key cryptosystems for the post-quantum period / Kuznetsov A., Svatovskij I., Kiyan N., Pushkar'ov A. 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T). Kharkov, 2017. pp. 125-130.
[20] Towards post-quantum security for IoT endpoints with NTRU / Guillen O. M. and etc. Design, Automation & Test in Europe Conference & Exhibition (DATE). Lausanne, 2017. pp. 698–703.
[21] Code-based key encapsulation mechanisms for post-quantum standardization / Kuznetsov A. and etc. 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT). Kyiv, 2018. pp. 276-281.
[22] Baldi M., Santini P., Cancellieri G. Post-quantum cryptography based on codes: State of the art and open challenges. AEIT International Annual Conference. Cagliari, 2017. pp. 1-6.
[23] Alam M. S. Secure M-commerce data using post quantum cryptography. 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI). Chennai, 2017. pp. 649-654.
[24] Post-Quantum Diffie-Hellman and Symmetric Key Exchange Protocols / Xiangdong Li and etc. 2006 IEEE Information Assurance Workshop. NY: West Point, 2006. pp. 382-383.

Рецензент: Іван Горбенко, д.т.н., професор, академік Академії наук Прикладної Радіоелектроніки, Харківський національний університет імені В.Н. Каразіна, майдан Свободи 6, м. Харків, 61022, Україна.
E-mail: gorbenkoi@iit.kharkov.ua

Надійшло: Вересень 2018.

Автори:

Олексій Нарежний, к.т.н., доцент факультету комп'ютерних наук, Харківський національний університет ім. В.Н. Каразіна, майдан Свободи 6, м. Харків, 61022, Україна.
E-mail: o.nariezhnii@karazin.ua

Егор Еремин, студент факультету комп'ютерних наук, Харківський національний університет ім. В.Н. Каразіна, майдан Свободи 6, м. Харків, 61022, Україна.
E-mail: suvenick2@gmail.com

Владислав Фроленко, студент факультету комп'ютерних наук, Харківський національний університет ім. В.Н. Каразіна, майдан Свободи 6, м. Харків, 61022, Україна.
E-mail: jadson27101@gmail.com

Кирилл Чернов, студент факультету комп'ютерних наук, Харківський національний університет ім. В.Н. Каразіна, майдан Свободи 6, м. Харків, 61022, Україна.
E-mail: kirillfilippsky@gmail.com

Татьяна Кузнецова, студентка факультету комп'ютерних наук, Харківський національний університет ім. В.Н. Каразіна, майдан Свободи 6, м. Харків, 61022, Україна.
E-mail: kuznetsova.tatiana17@gmail.com

Евгений Деменко, студент факультету комп'ютерних наук, Харківський національний університет ім. В.Н. Каразіна, майдан Свободи 6, м. Харків, 61022, Україна.
E-mail: demenjay@gmail.com

Статистичні властивості сучасних потокових шифрів.

Анотація. В останні роки в Україні продовжуються численні дослідження поточних симетричних шифрів, основною метою яких є аргументація принципів створення нового криптографічного алгоритму, на основі якого може бути прийнято національний стандарт. Одним з найважливіших аспектів вибору з багатьох варіантів є статистичні властивості вихідної псевдовипадкової послідовності (потоку ключів). У даній роботі отримані результати порівняльних досліджень статистичних властивостей вихідних послідовностей, які формуються різними потоковими шифрами, зокрема, всесвітньо відомі алгоритми Enocoro, Decim, Grain, HC, MUGI, Mickey, Rabbit, RC-4, Salsa20, SNOW2.0, Sosemanuk, Trivium та український криптографічний алгоритм Струмок, розроблений в останні роки. Для порівняльних досліджень була використана методика NIST

STS, згідно з якою експериментальні дослідження виконуються в 15 статистичних тестах, метою яких є визначення випадковості вихідних послідовностей. Кожен з цих тестів спрямований на вивчення певних вразливостей генератора, тобто вказує на потенційне використання різних методів криптографічного аналізу. Незважаючи на те, що розглянуті поточні алгоритми шифрування були вже досліджені раніше, ми провели статистичну перевірку сгенерованих послідовностей при однакових умовах і з однаковими початковими параметрами, тобто наші результати дозволяють провести порівняльний аналіз шифрів. Оцінки, представлені у статті, як і очікувалося, підтвердили високі показники статистичної безпеки сучасних шифрів. Крім того, за результатами експериментальних досліджень було встановлено, що новий український поточний шифр Струмок не поступається кращим світовим алгоритмам за статистичними властивостями послідовностей.

Ключові слова: симетрична криптографія; потокові криптографічні алгоритми; гамма; криптоаналіз; статистичні випробування.

Рецензент: Иван Горбенко, д.т.н., профессор, академик Академии наук Прикладной Радиоэлектроники, Харьковский национальный университет имени В. Н. Каразина, пл. Свободы 6, 61022, г. Харьков, Украина.
E-mail: gorbenkoi@iit.kharkov.ua

Поступила: Сентябрь 2018.

Авторы:

Алексей Нарежный, к.т.н., доцент факультета компьютерных наук, Харьковский национальный университет имени В.Н. Каразина, пл. Свободы 6, г. Харьков, 61022, Украина.
E-mail: o.nariezhnii@karazin.ua

Егор Еремин, студент факультета компьютерных наук, Харьковский национальный университет имени В.Н. Каразина, пл. Свободы 6, г. Харьков, 61022, Украина.
E-mail: suvenick2@gmail.com

Владислав Фроленко, студент факультета компьютерных наук, Харьковский национальный университет имени В.Н. Каразина, пл. Свободы 6, г. Харьков, 61022, Украина.
E-mail: jadson27101@gmail.com

Кирилл Чернов, студент факультета компьютерных наук, Харьковский национальный университет имени В.Н. Каразина, пл. Свободы 6, г. Харьков, 61022, Украина.
E-mail: kirillfilippsky@gmail.com

Татьяна Кузнецова, студентка факультета компьютерных наук, Харьковский национальный университет имени В.Н. Каразина, пл. Свободы 6, г. Харьков, 61022, Украина.
E-mail: kuznetsova.tatiana17@gmail.com

Евгений Деменко, студент факультета компьютерных наук, Харьковский национальный университет имени В.Н. Каразина, пл. Свободы 6, г. Харьков, 61022, Украина.
E-mail: demenjay@gmail.com

Статистические свойства современных поточных шифров.

Аннотация. В последние годы в Украине продолжаются многочисленные исследования потоковых симметричных шифров, основной целью которых является аргументация принципов создания нового криптографического алгоритма, на котором может базироваться национальный стандарт. Одним из важнейших аспектов выбора из многих вариантов являются статистические свойства псевдослучайной последовательности (потока ключей). В данной работе получены результаты сравнительных исследований статистических свойств последовательностей, которые формируются различными потоковыми шифрами, в частности, рассмотрены всемирно известные алгоритмы Enocoro, Decim, Grain, HC, MUGI, Mickey, Rabbit, RC-4, Salsa20, SNOW2 .0, Sosemanuk, Trivium и украинский алгоритм Струмок, разработанный в последние годы. Для сравнительных исследований была использована методика NIST STS, согласно которой экспериментальные исследования выполняются по 15 статистическим тестам, целью которых является определение случайности последовательностей. Каждый из этих тестов направлен на изучение определенных уязвимостей генератора, то есть указывает на потенциальное использование различных методов криптографического анализа. Несмотря на то, что рассмотренные алгоритмы шифрования были уже исследованы ранее, мы провели статистическую проверку сгенерированных последовательностей при одинаковых условиях и с одинаковыми начальными параметрами, то есть наши результаты позволяют провести сравнительный анализ шифров. Оценки, представленные в статье, как и ожидалось, подтвердили высокие показатели статистической безопасности современных шифров. Кроме того, по результатам экспериментальных исследований было установлено, что новый украинский поточный шифр Струмок не уступает лучшим мировым алгоритмам по статистическим свойствам последовательностей.

Ключевые слова: симметричная криптография; потоковые криптографические алгоритмы; гамма; криптоанализ; статистические тесты.

UDC 004.056.55

TESTING THE SPEED OF MODERN STREAM CIPHERS

Ivan Gorbenko[1], Yurii Gorbenko[1], Vladyslav Tymchenko[1], Olena Kachko[2]

[1] V. N. Karazin Kharkiv National University, 6 Svobody Sq., Kharkiv, 61022, Ukraine
gorbenkoi@iit.kharkov.ua, gorbenkou@iit.kharkov.ua, tvlad.tyma@gmail.com

[2] Kharkiv national university of Radio Electronics, 14 Nauky Av., Kharkiv, 61000, Ukraine
iit@iit.com

Reviewer: Vyacheslav Kalashnikov, Doctor of Sciences (Physics and Mathematics), Full Prof., Department of Systems and Industrial Engineering, Campus Monterrey, Monterrey, 64849, Mexico.
kalash@itesm.mx

Received on September 2018

Abstract. *The paper is a continuation of numerous studies of a candidate for encryption national standard of Ukraine, the Strumok new symmetric stream cipher. The result of a study is compare the most famous algorithms of stream cipher, which were presented at various contests, such as eSTREAM, NESSIE and the AES symmetric block cipher in mode of stream about usage CPU time for transformation of one octet data*.

Keywords: *stream cipher; encryption; cycles per byte; synchronous keystream generator; pseudorandom sequence*.

1 Introduction

Strumok is the stream symmetric cipher that presented as a candidate for encryption national standard of Ukraine [1-4]. The cipher designed by a group of scientists in 2015 and it was improve repeatedly. The Strumok has a simple scheme that focuses on a 64-bit computing platform, that enable its become the most fast (over 10 Gbit per sec) [1-4].

The Strumok stream cipher is based on a secret internal state that consist of eighteen 64-bit blocks: 16 blocks of the linear feedback shift register (LFSR) and two blocks of the finite-state machine (FSM). As input data using a secret key of size L_K=256 or 512 bit and an initial vector of size L_{IV}=256 bit.

The closest prototype to the Strumok stream cipher is the SNOW 2.0 cryptographic algorithm that focuses on 32-bit computing systems [5]. It is based on the classical scheme of additive generator [5,6], but unlike the SNOW 2.0 algorithm in the Strumok generator uses an increased internal state with a shift register over 64-bit blocks. Therefore, in one cycle, 64-bit computing systems achieve a higher speed of the formation keystream.

The purpose of this work is obtaining additional results of compare studies speed the Strumok stream cipher and famous cryptographic primitives about usage CPU time for transformation of one octet data at various stages of keystream generation. By comparison, also attached the symmetric block ciphers, such as AES [7,8] and Kalyna [9,10] with usages in stream mode.

2 General Description

AES

Advanced Encryption Standard (AES) is a symmetric block cipher that has a fixed block size of 128 bit, and a key length can take value L_k=128,192 or 256 bit. For a key lengths 128 or 256 bit, the algorithm has 10/14 rounds, according. In 2002, it was declared the encryption standard of United States [7] and later its standardized at the international level in [8].

HC

HC is a stream cipher that designed by Hongjun Wu cryptographer and was first published in 2004. HC-128 was presented at the eSTREAM contest [6], which aimed to create European standards for stream encryption systems.

Rabbit

Rabbit is a high-performance stream encryption algorithm that was first presented at the 10th symposium FSE in 2003. In 2005, it was submitted at the eSTREAM contest [6]. The cipher uses a 128-bit key and a 64-bit initialization vector. It standardized internationally in [5].

Salsa20

Salsa20 is a transformation stream system that developed by Daniel J. Bernstein. The algorithm was presented at the eSTREAM contest [6], where it became the winner contest in the first profile (the stream ciphers for software use with high bandwidth).

SNOW 2.0

SNOW 2.0 is a symmetric stream cipher that developed by Thomas Johansson and Patrik Ekdahl, it's also one of the stream ciphers selected for ISO/IEC 18033-4 [5]. The key size can take 128 or 256 bit, and a 128-bit initialization vector.

Sosemanuk

Sosemank is a stream cipher, developed by a group of French scientists in 2004. In 2008, it became one of the finalists eSTREAM project in the first profile [6]. It's a key length varies between 128 and 256 bit, and use a 128-bit initialization vector.

Trivium

Trivium is a synchronous stream cipher that focuses primarily on hardware implementation, and reasonably efficient software implementation. The cipher was presented at the eSTREAM project and has been selected as part portfolio for low area hardware ciphers [6]. The authors of the cipher are Christophe De Canniere and Bart Prenel. This stream cipher can generate up to 2^{64} bits of output sequence with an 80-bit key and an 80-bit initialization vector. Standardized also as a cipher of lightweight cryptography [11].

3 Software performance of Strumok

The stream encryption of long sequences has the potential advantage over block cryptographic transformations, which is an important benchmark for many applications. Important benchmarks of symmetric stream ciphers are also the speed of encrypting short packages, as well as the time of initialization/generation of key parameters (see in Tables 1-10).

Table 1 – The speed of encryption 1Gb data (*Intel Core i3-5005U 2ГГц*)

Names Of Algorithm	***Speed***	
	Mbps	***Cycles per byte***
Strumok-256	5542	2.88
Strumok-512	5576	2.86
AES-128	1583	10.08
AES-256	1128	14.14
HC-128	7164	2.23
HC-256	2701	5.91
Rabbit	2186	7.3
Salsa20	1598	9.98
SNOW 2.0-128	5177	3.08
SNOW 2.0-256	5095	3.13
Sosemanuk	2925	5.46
Trivium	2387	6.69

To study the performance benchmarks used two computers:

- with an Intel® Core i3-5005U 2 GHz processor (*the memory cache: 128 KB first level and 256 KB second level*), 12 GB of DDR3 1600 MHz RAM and OS Windows® 8.1;

- Intel® Core i9-7980XE 2.60 GHz processor (*the memory cache: 18×32 KB first level and 1024 KB second level*), 64 GB of DDR4 2133 MHz RAM and OS Windows® 10 Pro.

Table 2 – The speed of encryption 1Gb data *(Intel Core i9-7980XE 2.60GHz)*

Names Of Algorithm	***Speed***	
	Mbps	***Cycles per byte***
Strumok-256	10911	1.90
Strumok-512	10850	1.91
AES-128	3228	6.42
AES-256	2295	9.03
HC-128	14676	1.41
HC-256	6286	3.30
Rabbit	4636	4.47
Salsa20	3502	5.92
SNOW 2.0-128	10425	1.99
SNOW 2.0-256	10470	1.98
Sosemanuk	6329	3.28
Trivium	4954	4.19

Table 3 – The results of encryption 50 packages by 1500 bytes *(Intel Core i3-5005U 2ГГц)*

Names Of Algorithm	***Speed***		
	Mbps	***Cycles per byte***	***Cycles per package***
Strumok-256	4580	3.49	5231
Strumok-512	4545	3.50	5250
AES-128	1578	10.13	15194
AES-256	1119	14.25	21368
HC-128	1310	12.18	18275
HC-256	239	66.57	99861
Rabbit	2105	7.59	11378
Salsa20	1578	10.11	15161
SNOW 2.0-128	4687	3.42	5127
SNOW 2.0-256	4580	3.48	5218
Sosemanuk	2830	5.65	8481
Trivium	2197	7.25	10876

Table 4 – The results of encryption 50 packages by 1500 bytes *(Intel Core i9-7980XE 2.60GHz)*

Names Of Algorithm	***Speed***		
	Mbps	***Cycles per byte***	***Cycles per package***
Strumok-256	9231	2.26	3386
Strumok-512	9091	2.26	3387
AES-128	3191	6.47	9710
AES-256	2299	9.03	13545
HC-128	2871	7.21	10819

Continuation of Table 4

HC-256	543	38.21	57318
Rabbit	4255	4.86	7287
Salsa20	3448	6.01	9008
SNOW 2.0-128	9375	2.20	3303
SNOW 2.0-256	9524	2.18	3274
Sosemanuk	7595	2.74	4108
Trivium	4511	4.59	6887

The research was conducted in accordance with the methodology adopted in [6], i.e., the following criteria were used:

- the speed of encryption of long streams;
- the speed of encryption short packages;
- the speed of initialization and key parameters generation.

The results obtained of encryption a long stream (a long 1GB) which carried out for each algorithm on a key, see in Tables 1, 2. From the data in tables, it follows that stream ciphers have an undeniable advantage over blocks when encryption long streams. Among the stream algorithms, HC-128, Strumok-256 and Strumok-512 are the fastest. It should be noted that Intel® Core i9-7980XE 2.60 GHz computing system achieves very high encryption speeds (*over 10 Gbit per sec*), which points to the prospect of using stream ciphers in modern telecommunication systems. At the same time, Strumok-512 is not inferior to the speed of the Strumok-256 version, although it has a much higher supply of stability.

Table 5 – The results of encryption 120 packages by 576 bytes *(Intel Core i3-5005U 2ГГц)*

Names Of Algorithm	***Speed***		
	Mbps	***Cycles per byte***	***Cycles per package***
Strumok-256	3477	4.59	2644
Strumok-512	3456	4.61	5250
AES-128	1562	10.20	5876
AES-256	1103	14.47	21368
HC-128	570	27.95	16099
HC-256	97	164.5	94753
Rabbit	1953	8.15	4696
Salsa20	1612	9.91	5707
SNOW 2.0-128	4189	3.80	2191
SNOW 2.0-256	3711	4.31	2484
Sosemanuk	2850	5.58	3217
Trivium	1920	8.33	4798

Table 6 – The results of encryption 120 packages by 576 bytes *(Intel Core i9-7980XE 2.60GHz)*

Names Of Algorithm	***Speed***		
	Mbps	***Cycles per byte***	***Cycles per package***
Strumok-256	6999	2.96	1702
Strumok-512	7089	2.94	1695

Continuation of Table 6

AES-128	3196	6.48	3732
AES-256	2257	9.18	5288
HC-128	1254	16.53	9520
HC-256	220	94.24	54284
Rabbit	4036	5.13	2957
Salsa20	3392	6.08	3503
SNOW 2.0-128	8378	2.44	1408
SNOW 2.0-256	8507	2.42	1396
Sosemanuk	6356	3.33	1919
Trivium	3921	5.30	3054

By the second criterion, the speed of encryption short packages of different lengths was measured. Each function call includes a separate setup of the initialization vector *IV*. The package lengths (40, 576, and 1500 bytes) were selected to be representative of telecommunication traffic (see Table 3-8) [6].

Analyses the results of the studies presented in Tables 3–8, it should be noted that the advantage the speed of encryption stream algorithms is maintained for packages of lengths hundred or more bytes. For very short packages, the time of internal state initialization begins to play a significant part of the stream algorithms, as expected, are starting to lose. As regards comparing the speed of stream algorithms, the advantage of the SNOW 2.0 and Strumok generators should be noted.

Table 7 – The results of encryption 350 packages by 40 bytes *(Intel Core i3-5005U 2ГГц)*

Names Of Algorithm	***Speed***		
	Mbps	***Cycles per byte***	***Cycles per package***
Strumok-256	543	29.32	1173
Strumok-512	528	30.21	1209
AES-128	1027	15.63	625
AES-256	713	22.38	895
HC-128	42	378.39	15135
HC-256	7	2296.64	91866
Rabbit	682	23.39	936
Salsa20	1009	15.7	628
SNOW 2.0-128	888	17.83	713
SNOW 2.0-256	957	16.67	667
Sosemanuk	756	21.08	843
Trivium	493	32.38	1295

Table 8 – The results of encryption 350 packages by 40 bytes *(Intel Core i9-7980XE 2.60GHz)*

Names Of Algorithm	***Speed***		
	Mbps	***Cycles per byte***	***Cycles per package***
Strumok-256	1120	18.45	738
Strumok-512	1131	18.23	729

Continuation of Table 8

AES-128	2196	9.48	379
AES-256	1600	12.87	515
HC-128	94	219.82	8793
HC-256	16	1326.64	53065
Rabbit	1623	13.04	522
Salsa20	2153	9.70	388
SNOW 2.0-128	2154	9.67	387
SNOW 2.0-256	2154	9.69	388
Sosemanuk	1697	12.28	491
Trivium	1028	20.14	806

The criterion of initialization and generation key parameters separately includes parameters of the establishment of the key and the initialization vector. These two parameters are least critical for displaying the speed of encryption packages, since they are disproportionality small compared to the process of create or update a key. The results benchmarks are shown in the Tables 9, 10.

According to the results of the studies presented in Table 9, 10, the advantage of the Salsa20 cipher should be noted. The HC algorithm, which showed good results the speed, it has the shortest time of installation a key, but the time it takes to initialize the vector is largest. The Strumok algorithm has average values for this criterion.

Table 9 – The speed OF the installation keys and the initialization vectors
(Intel Core i3-5005U 2ГГц)

Names Of Algorithm	**KEY, Cycles per installation**	**IV, Cycles per installation**
Strumok-256	16.07	806.52
Strumok-512	16.16	731.24
AES-128	266.35	0.32
AES-256	423.39	0.32
HC-128	8.07	14701.36
HC-256	133.07	91748.16
Rabbit	475.35	442.69
Salsa20	12.08	1.59
SNOW 2.0-128	16.09	403.21
SNOW 2.0-256	30.21	410.64
Sosemanuk	530.39	496.41
Trivium	22.35	1006.43

Table 10 – The speed OF the installation keys and the initialization vectors
(Intel Core i9-7980XE 2.60GHz)

Names Of Algorithm	**KEY, Cycles per installation**	**IV, Cycles per installation**
Strumok-256	10.24	470.32
Strumok-512	10.24	471.67
AES-128	152.02	0.15

Continuation of Table 10

AES-256	242.14	0.16
HC-128	7.08	8640.58
HC-256	79.44	52514.03
Rabbit	285.73	275.32
Salsa 20	7.69	0.98
SNOW 2.0-128	10.38	238.89
SNOW 2.0-256	19.30	233.66
Sosemanuk	339.99	317.19
Trivium	14.11	617.50

4 Conclusions

The symmetric stream ciphers play an important role in the processes cryptographic protection of information. They have high the speed of cryptographic transformation, especially for the long packages, and are most successfully used when encryption large amounts of input data.

The results obtained of comparative studies have shown that the stream algorithms significantly exceed block ciphers by the speed of encryption the long packages. Among the stream algorithms, the Strumok generator, whose structure is targeted at applications in modern 64-bit computing system, has the advantage of giving it the highest values. In particular, the Intel® Core i9-7980XE 2.60 GHz and OS Windows® 10 Pro have received 14 – 15 Gbps encryption speeds.

When encryption short packages, the computational efficiency of the stream ciphers decreases, and for package lengths of 40 bytes block ciphers become faster. When comparing the stream algorithms, the Strumok generator that stably shows a high encryption speeds, has the advantage.

The study of the initialization time of cryptographic algorithms didn't show the advantage of block or stream algorithms. And although with the increase in the length of the data being processed, the initialization time plays a very small part in the process of encryption, this parameter should also to be given attention. In particular, according to our research, the greatest advantage over the initialization time is the Salsa20 cipher.

Generalizing the results should be noted the Strumok keystream generator, which in most cases showed better results. When implemented on a 64-bit computing platform, it provides enormous the speed of encryption and can be recommended for practical application in the modern information and telecommunication systems.

References

[1] Kuznetsov O., Lutsenko M., Ivanenko D. Strumok stream cipher: Specification and basic properties. 2016 Third International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T). Kharkiv, 2016. pp. 59–62.

[2] The research of modern stream ciphers / Gorbenko I., Kuznetsov A., Lutsenko M., Ivanenko D. 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T). Kharkov, 2017. pp. 207–210.

[3] Strumok keystream generator / Gorbenko I. and etc. 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT). 2018. pp. 294–299.

[4] Kuznetsov A., Frolenko V., Eremin E., Zavgorodnia O. Research of cross-platform stream symmetric ciphers implementation. 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT). Kyiv, 2018. pp. 300–305.

[5] ISO/IEC 18033-4:2011. Information Technology-Security Techniques-Encryption Algorithms-Part 4: Stream ciphers. URL: http://www.iso.org/iso/home/store/catalogue_ics/ catalogue_detail_ics.htm?csnumber=54532 [Dec., 2012].

[6] The eSTREAM Project. URL: http://www.ecrypt.eu.org/

[7] FIPS-197: Advanced Encryption Standard (AES). NIST, 2001. URL: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[8] ISO/IEC 18033-3:2010. Information technology – Security techniques – Encryption algorithms – Part 3: Block ciphers. URL: https://www.iso.org/standard/54531.html

[9] A New Encryption Standard of Ukraine: The Kalyna Block Cipher. URL: https://eprint.iacr.org/2015/650.pdf

[10] DSTU 7624:2014. Information technologies. Cryptographic Data Security. Symmetric block transformation algorithm. URL: http://shop.uas.org.ua/ua/catalogsearch/result/?q=7624 (in Ukrainian).

[11] ISO/IEC 29192-3:2012. Information technology-Security techniques-Lightweight cryptography-Part 3: Stream ciphers. URL: https://www.iso.org/standard/56426.html

Рецензент: В'ячеслав Калашников, д.ф.-м.н., проф., Технологический університет Монтеррея, пр. Еухеніо Гарса Сада 2501, Монтеррей, 64849, Мексика. E-mail: kalash@itesm.mx

Надійшло: Вересень 2018.

Автори:

Іван Горбенко, д.т.н., проф., академік Академії наук прикладної радіоелектроніки, Харківський національний університет ім. В.Н. Каразіна, майдан Свободи 6, м. Харків, 61022, Україна. E-mail: gorbenkoi@iit.kharkov.ua

Юрій Горбенко, к.т.н, Харківський національний університет ім. В.Н. Каразіна, майдан Свободи 6, м. Харків, 61022, Україна. E-mail: gorbenkou@iit.kharkov.ua

Владислав Тімченко, студент факультету комп'ютерних наук, Харківський національний університет ім. В.Н. Каразіна, майдан Свободи 6, м. Харків, 61022, Україна. E-mail: tvlad.tyma@gmail.com

Олена Качко, к.т.н., проф., Харківський національний університет радіоелектроніки (ХНУРЕ), м. Харків, 61000, Україна. E-mail: iit@iit.com.ua

Дослідження швидкості сучасних потокових шифрів.

Анотація. Ця стаття є продовженням численних досліджень кандидата на національний стандарт шифрування України, нового алгоритму симетричного потокового шифрування «Струмок». Результатом дослідження є порівняння найвідоміших алгоритмів потокового шифрування, які були представлені на різних конкурсах, таких як eSTREAM, NESSIE. До порівняння також долучений симетричний блоковий шифр AES в потокових режимах застосування. Оцінювалася складність реалізації алгоритмів за показниками кількості циклів центрального процесора для перетворення одного октету даних.

Ключові слова: потоковий шифр; шифрування; цикли на байт; синхронний генератор ключового потоку; псевдовипадкова послідовність.

Рецензент: Вячеслав Калашников, д.т.н., проф., Технологический университет Монтеррея, пр. Еухенио Гарса Сада 2501, Монтеррей, 64849, Мексика. E-mail: kalash@itesm.mx

Поступила: Сентябрь 2018.

Авторы:

Иван Горбенко, д.т.н., проф., Академик Академии наук прикладной радиоэлектроники, Харьковский национальный университет имени В.Н. Каразина, пл. Свободы 6, г. Харьков, 61022, Украина. E-mail: gorbenkoi@iit.kharkov.ua

Юрий Горбенко, к.т.н. Харьковский национальный университет имени В.Н. Каразина, пл. Свободы 6, г. Харьков, 61022, Украина. E-mail: gorbenkou@iit.kharkov.ua

Владислав Тимченко, студент факультета компьютерных наук, Харьковский национальный университет имени В.Н. Каразина, пл. Свободы 6, г. Харьков 61022, Украина. E-mail: tvlad.tyma@gmail.com

Елена Качко, к.т.н., проф., Харьковский национальный университет радиоэлектроники (ХНУРЭ), г. Харьков, 61000, Украина. E-mail: iit@iit.com.ua

Исследование скорости современных поточных шифров.

Аннотация. Эта статья является продолжением многочисленных исследований кандидата на национальный стандарт шифрования Украины, нового алгоритма симметричного поточного шифрования «Струмок». Результатом исследования является сравнение наиболее известных алгоритмов потокового шифрования, которые были представлены на различных конкурсах, таких как eSTREAM, NESSIE. В сравнение также включен симметричный блочный шифр AES в потоковых режимах применения. Оценивалась сложность реализации алгоритмов по показателям количества циклов центрального процессора для преобразования одного октета данных.

Ключевые слова: потоковый шифр; шифрование; циклы на байт; синхронный генератор ключевого потока; псевдослучайная последовательность.

Статті пройшли внутрішнє та зовнішнє рецензування.

Наукове видання

КОМП'ЮТЕРНІ НАУКИ ТА КІБЕРБЕЗПЕКА

Випуск 3(11) 2018

Міжнародний електронний науково-теоретичний журнал

Англійською, українською, російською мовами

Комп'ютерне верстання – Єсіна М.В., Федоренко В.В.

61022, Харків, майдан Свободи, 6
Харківський національний університет імені В.Н. Каразіна

V. N. Karazin Kharkiv National University Publishing

CS&CS