ISSN 2519-2310

CS&CS Journal

KARAZIN UNIVERSITY
CLASSICS AHEAD OF TIME

1(13) 2019

COMPUTER SCIENCE AND CYBERSECURITY

КОМП'ЮТЕРНІ НАУКИ ТА КІБЕРБЕЗПЕКА

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 1(13) 2019

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 (June 24, 2019, protocol No. 7)

The journal has Digital Object Identifier: **10.26565/2519-2310**.

Editorial office:
V.N. 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 1(13) 2019

UDC 004.056.55 DOI: 10.26565/2519-2310-2019-1-01

MATHEMATICAL MODEL OF THE BIOMETRIC SYSTEM OF FINGERPRINT AUTHENTICATION

Serhii Rassomakhin[1], Kateryna Budianska[1], Anna Uvarova[2], Mykhaylo Bagmut[1]

[1] V. N. Karazin Kharkiv National University, Svobody Sq., 4, Kharkiv, 61022, Ukraine
rassomakhin@karazin.ua, budyanskaya96@gmail.com, sapsanmiha@gmail.com

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

Reviewer: Serhii Toliupa, Doctor of Sciences (Engineering), Full Prof., Taras Shevchenko National University of Kiev, 81 Lomonosova St., Kyiv, 03189, Ukraine.
tolupa@i.ua

Received on February 2019

Abstract: *This paper considers mathematical models of biometric fingerprint images, as well as basic computational procedures for fingerprinting. The main stages of processing dactyloscopic portraits based on the selection of local features, their filtering and digital processing are investigated. The developed software implements the transformation of fingerprint images with the subsequent formation of a cryptographically strong password sequence based on them. This allows you to simulate a dactyloscopic authentication system for the purpose of studying certain of its properties, estimating probabilistic performance indicators (error probabilities of the first and second kind), and so on.*

Keywords: *fingerprints; biometric image; password authentication; biometric system.*

1 Introduction

To date, biometric technologies are actively used in many areas of everyday life that are related to providing access to information in the tasks of identification and authentication of personality. Biometrics use the following features that are inherent to each individual: a papillary finger pattern, a pattern of the iris, voice parameters, a blood vessel pattern and so on.

All people have a unique fingerprint pattern, so everyone can be identified. Identification algorithms use points on fingerprints: the end of the pattern line, line branching, single points [1]. Also, the morphological structure of the fingerprint is considered, namely, the relative position of the closed, arched and spiral lines of the papillary pattern. The features of fingerprint portraits are transformed into a unique code that preserves the informative image of the imprint [1].

Thus, the actual formation of a model of a biometric fingerprint portraiture system and subsequent authentication is relevant. The purpose of this work is to develop a mathematical method and algorithm for converting fingerprint portraits for reliable authentication on fingerprints.

2 Formation of the model of fingerprints

Dactyloscopy is a subdiscipline of traosology that studies the configuration of papillary lines of the skin on the palms, fingers, legs and special techniques for their investigation in order to establish the identity of the person in the process of identification and authentication of users for access to a secure system, etc. [2]. There are two types of attributes: global and local. The global features include the core (center), the point "delta" (*starting point*), line type, counter line, types of patterns. Local signs are also minutias that are unique for each imprint of signs that determine the points of change in the structure of papillary lines (ending, splitting, rupture, etc.), the orientation of the papillary lines and the coordinates in these points. Each imprint can contain from 16 to 70 minutias [3].

Classification of all fingerprints is based on the existence of singular points - global fingerprint signs [4]. Although the fingerprints of different people may have the same global characteristics, it is completely impossible to have the same local characteristics. Therefore, global attributes are used to

classify the database into classes - the authentication phase. At the next stage, local identification is used to identify.

Papillary patterns on the human fingertips form three types of patterns, namely: "*loop*" (left, right, central, double), *delta* or arc (simple and acute), "*spiral*" (central and mixed) (Fig. 1).

a) *b)* *c)*

Fig. 1 - Types of fingerprint patterns (*a – loop; b – delta; c - spiral*)

The fundamental distinction between identification and authentication is the level of trust to the user. At the previous stage of system identification, the level of trust to the user being logged is a priori high. In a multi-user system, biometric identification must be carried out under the direct control of its owner or its representative, which confirms the user's authority and correctness of the behavior during the system training.

The biometric authentication mode, conversely, implies a low level of confidence in the identity that is authenticated. When biometric authentication, the applicant must prove the authenticity of his claimed name by presenting his unique biometric images. It should be noted that biometric authentication is potentially vulnerable if it is used regardless of the methods of classical authentication based on protocols using passwords and keys. An adequate level of information security can only be achieved by combining methods of classical and biometric authentication.

Any biometric system can accept errors of the first and the second kind. In biometrics, the most constant notions are FAR (*False Acceptance Rate*) and FRR (*False Rejection Rate*). The first number characterizes the probability of a false coincidence of the biometric characteristics of two people. The second is the likelihood of denial of access to a person with a tolerance. The system is considered to be better when the value of FRR is smaller with the same FAR values [5]. Fingerprint has a characteristic FAR of 0.001%, and FRR of 0.6% [6].

As matters currently stand, it can be identified three basic methods for comparing fingerprint images: comparing patterns of prints, correlation comparisons and a method that uses key points - minutias.

In the algorithm of pattern comparison, the peculiarities of the structure of the papillary pattern are used directly. The image of the fingerprint obtained from the scanner is divided into a large number of small particles, while the size of such cells depends on the required accuracy (the smaller the size of the cell, the more accurate the result is obtained).

The disposition of the lines in the cell is described by parameters of some sinusoidal wave: the initial phase shift, the wavelength and the direction of its propagation are determined. Accordingly, to receiving a fingerprint for comparison, it is aligned and is reduced to the same kind as the template. Then the parameters of the wave representations of the corresponding cells are compared.

The main advantages of this algorithm are quite high speed and low requirements for the quality of the resulting image. The method of comparison on the pattern has not yet become widespread because of the complexity of implementation, as well as high requirements to the mathematical base.

It should be noted separately that in the automated identification, there are several problems associated with the difficulty of scanning and recognizing some types of fingerprints.

In the algorithm that uses a correlation comparison, the received fingerprint is superimposed on each of the standards of the prints of the database and the calculation of the pixel difference between the input fingerprint and the reference.

The main advantage of this method is the low requirement for the quality of the received imprint. Disadvantages: the need for a large amount of memory to store the database, low speed algorithm. Every time a person places his finger at different angles and different places of the scanner's working area. This means that the process of comparing its fingerprint with the standards should include a lot of iterations, each of which the image obtained from the scanner returns at a small angle or slightly shifts [7]. Because of the duration of the comparison procedure, especially when solving the identification problem, the "one to many" comparison, this method is extremely rarely used for solving identification and authentication tasks.

The algorithm of a method that uses minutias describes a template that is formed on the basis of the fingerprint image on which the end points and branch points are marked. When comparing the key points at the input fingerprint image are also marked. After that, the minutias of the given imprint are compared with the templates. By the number of concurrency points, a decision is made on the identity of the images [8]. The advantage of this algorithm is the high speed of operation. That is why the algorithms of this class are the most common.

To work the algorithm for comparing fingerprints from key points, high quality images with low noise are required. Therefore, special image processing algorithms are used to improve the quality of the fingerprint images. In particular, as a model of noisy biometric imaging, the most commonly used normal distribution law with random variables. In this paper, the method of taking the inverse function is used to implement this model. This method is especially convenient to use in the case when the integral law of probability distribution is given analytically and a possible analytical take of the inverse function from it.

3 Modeling of normally distributed random variables in the processing of biometric images

The random variable means the value, which as a result of the trial takes a certain value, and it is unknown in advance, which is exactly [9].

According to the central limit theorem of probability theory, by adding a sufficiently large number of equally distributed independent random variables, we obtain a random variable with a normal distribution law [10] (1).

$$F\left(x\right) = P(X < x) = \sum_{x_i < x} P(X = x_i) = \sum_{x_i < x} p_i \,. \tag{1}$$

Due to the addition of more than ten random variables with uniform distribution in the interval $(0;1)$, we obtain a random variable, which with the accuracy sufficient for most practical problems can be considered as distributed according to the normal law [10].

For the quantitative characterization of the distribution law it is convenient to use the probability of an event $X < x$, where x- a certain variable. The probability of this event obviously depends on x. This dependence is given by the distribution function of a random variable X (2):

$$F(x) = P(X < x). \tag{2}$$

Properties of the distribution function are:

- the distribution function $F(x)$ does not decrease, that is, when $x_2 < x_1$ the $F(x_2) \geq F(x_1)$ inequality is performed;
- $F(-\infty) = 0$;
- $F(+\infty) = 1$.

Knowing the law of the distribution of random variables, we can construct the distribution function by the following rule.

The function of the distribution of a random variable always is a discontinuous step function, the jumps of which occur at points corresponding to the possible values of the random variable, and are equal to the probabilities of these values [11]. The sum of all jumps is equal to one (Fig. 2).

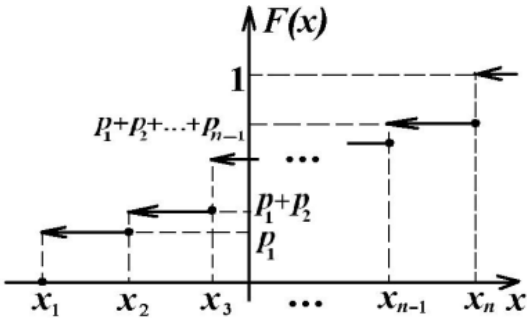


Fig. 2 - Bursting step function

Normal distribution law is found in nature very often, therefore, effective methods of modeling for it are developed. The formula for the probability distribution of the values of random variable x under the normal law has the form (3):

$$y = \frac{1}{\sigma_x \cdot \sqrt{2\pi}} \cdot e^{-\frac{(x-m_x)^2}{2\sigma_x^2}}, \quad (3)$$

Where x - a random variable; $y(x)$ - probability of acceptance of a random value of value x; m_x - mathematical expectation; σ_x - mean square deviation.

As you can see, a normal distribution has two parameters: the mathematical expectation m_x and the mean square deviation σ_x of the value x from this mathematical expectation.

Normalized normal distribution (Fig. 3) is called a normal distribution, which has $m_x = 0$ and $\sigma_x = 1$ [12]. With a normalized distribution, you can get any other normal distribution with given m_x and σ_x by the formula: $z = m_x + x \cdot \sigma_x$.

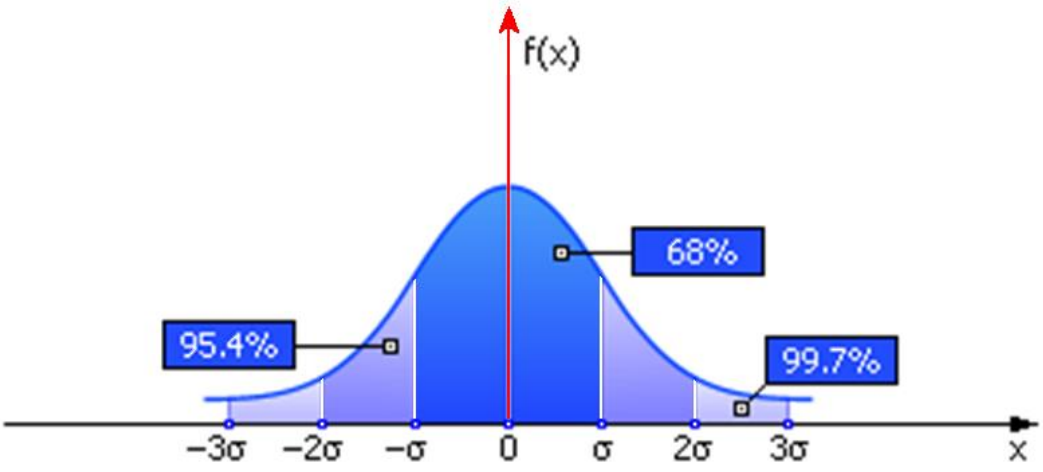


Fig. 3 - A graphical representation of the normal distribution law of a random variable x

The graph of the normal distribution law shows that in the interval $-\sigma < x < \sigma$ of the graph, 68% of the distribution area is concentrated, in the interval $-2\sigma < x < 2\sigma$ 95.4% of the distribution area is concentrated, and in the interval $-3\sigma < x < 3\sigma$ - 99.7% of the distribution area (the "rule of three sigmas").

Consider the question of simulating random variables given by the normal distribution law. In this paper, the method of taking the inverse function will be used.

In the case of continuous random variables, their probabilistic characteristics are determined by the distribution density. The density of the distribution of a random variable [13] X is called a function $f(x)$ that (4):

$$f(x) = F'(x), \tag{4}$$

where $F(x)$ is the distribution function of the quantity X.

Assume that the integral law of probability distribution $F(x)$ is given to us (5):

$$F(x) = \int_{-\infty}^{x} f(x)\,dx. \tag{5}$$

Then it is enough to play a random number, evenly distributed in the range from 0 to 1. Since the function F also varies in this interval, then the random event x can be determined by taking a reciprocal function graphically or analytically: $x = F^{-1}(r)$. Here r - the number generated by the reference *Random Number Generator* in the range from 0 to 1, x - generated as a result a random variable. Graphically, the essence of the method is depicted in Fig. 4, namely the probability density graphs and the integral probability density of x.

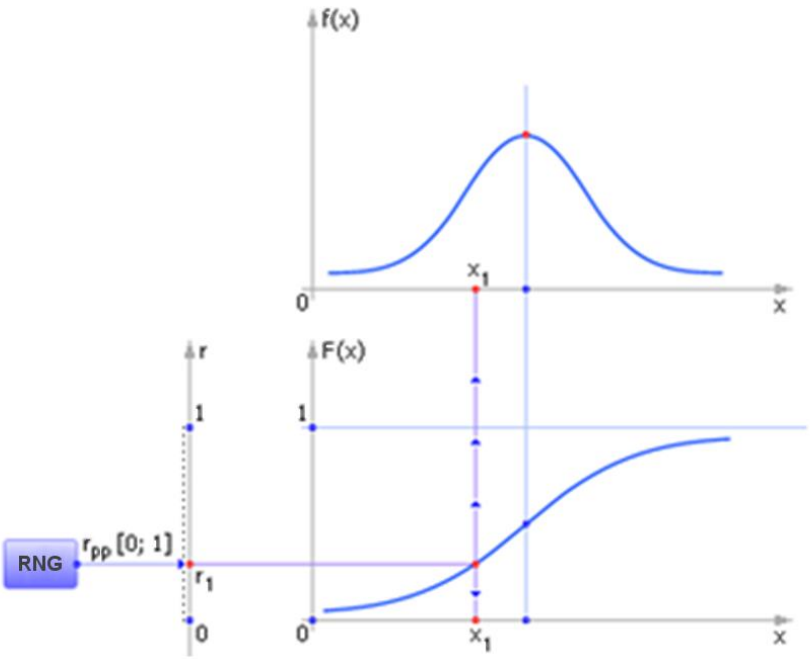


Fig. 4 - Illustration of the reverse function method for generating random events x

By the density distribution property $f(x) > 0$ for all x of the interval (a,b), the function $y = F(x)$ increases strictly in this interval (Fig. 3). Therefore, the function $F(x)$ has an unambiguous inverse function $x = G(y)$ with a range of values $y \in (0,1)$ and a definition area $x \in (a,b)$ on the interval (a,b). The function $x = G(y)$, as well as the function $y = F(x)$, is increasing (Fig. 5).

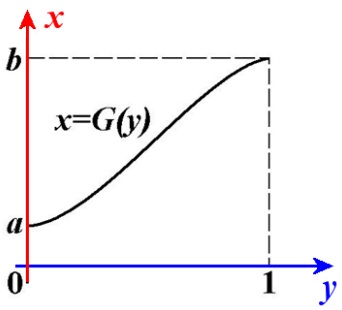


Fig. 5 - Graphic display of the function $x = G(y)$

This function also has an unambiguous reverse function on its interval of definition $(0,1)$. Obviously, this reverse function is $y = F(x)$. This method is especially convenient to use in the case when the integral law of probability distribution is given analytically and a possible analytical take of the inverse function from it.

4 Handles dactyloscopic portraits

4.1 Selection of key points in the image

Image processing is an important part in creating automated biometric identification systems.

Usually, the image obtained with the scanner is of low quality. The quality of this image is influenced by many factors: the strength of the finger pressure when scanning, the humidity of the air, the cleanliness of the finger and the scanner itself. Therefore, different methods of filtering are used to improve the quality of the original image. At the stage of thinning operation, the binary image of the image of the fingerprint is throttled to the skeleton.

The skeleton of the line is called a simple chain *(u,v)* passing near the geometric center of the line, and for each vertex p_1 there are exactly two adjacent to it vertices p_2 and p_3, thus, the vertices p_2 and p_3 are not adjacent.

Thinning operation is a morphological operation that brings a binary image to its skeleton. The thickness of all lines in the skeleton has a thickness of 1-2 pixels. For such a result, it is necessary to remove the extreme points from the black lines. The method pulls the line into the center, without making any breaks.

After that, in the skeleton of the image is searched for minutias - key points. In most algorithms for recognition, only two types of minutias are used: ending and branching.

The ending is called the top of the skeleton such that for it there is exactly one vertex adjacent to it. The branching is called a vertex of the skeleton, for which there are exactly three adjacent vertices and are pairwise noncontiguous.

For each key point, its type, their coordinates in the image and the orientation are determined.

To correctly determine the minutias, it is needed to process the image and then lead to a special format. The image processing can take place in two scenarios. Such scenarios are conditional, that is, it is possible to combine them.

The first scenario:

1) calculation of orientation of lines;
2) improving the quality of the lines;
3) binaryization of the image;
4) thinning operation to the image.

Second scenario:

1) adaptive filtering, allocation of interest zone;
2) binarization, the allocation of homogeneous areas;
3) morphological treatment;
4) thinning;
5) vectorization;
6) vector post-processing.

4.2 Digital fingerprint image processing

To date, there are several fingerprint filtering algorithms.

Smoothing filter

The smoothing filter is widely used to remove noise in the image in general and in the form of a fingerprint in particular. It consists in scanning the entire image of the window N - dimensionality $n*n$ and converting the intensity value for each pixel. The new value is calculated as the arithmetic mean of the value of all the pixels that fall into the window (6):

$$I(i,j) = \frac{\sum_{(x,y)}^{N} I(x,y)}{n^2}, \quad (6)$$

where $I(i,j)$ - the new value of the pixel intensity with coordinates (i,j), $I(x,y)$ - the initial intensity value for the pixel with coordinates (x,y).

Median filter.

Similarly, the method of removing noise in an image is widespread. The image is scanned by the $n \times n$ dimension window, the value of the intensity of the pixels inside each window is sorted ascending (descending); the output value is the intensity of the pixel located in the middle of the list.

The method of spatial filtration of the image.

The method of spatial filtration of the image is to realize the physical process of absorption and reflection of light. The algorithm is implemented in several stages:

1) The input of the algorithm receives a grayscale $n \times n$. Then the threshold processing of the fingerprint image is performed to obtain a binary image.

Let $R(i,j)$ - the value of the pixel with coordinates (i,j) in the binary image, then $R(i,j) = 1$ with $G(i,j) > R_0$, in all other cases $R(i,j) = 0$.

2) At this stage, the binary image is scanned by the $n \times n$ dimension window, for the central pixel of the window with coordinates (i,j), and the reflection coefficient is calculated, which is equal to the ratio of the number of $N(i,j)$ white pixels caught in the window to the dimension of the window (7):

$$k(i,j) = \frac{N(i,j)}{n^2}, \quad (7)$$

where $k(i,j)$ - the reflection coefficient of the pixel with coordinates (i,j).

3) Next, a new intensity value for each pixel is calculated. The new value of the intensity of the pixel is equal to the product of the reflection coefficient on the maximum intensity of light (8):

$$I(i,j) = k(i,j) \cdot I_{max}, \quad (8)$$

where I_{max} - the maximum value of intensity, $I(i,j)$ - the value of the intensity of the pixel with coordinates (i,j).

Gabor filter.

Processing of the image of a fingerprint by the given algorithm is carried out in several stages:

1) Normalize the image. Required to set the previous mean values and deviations. A normalized image G is defined as an image where $G(i,j)$ - the value of the normalized brightness of the pixel with coordinates (i,j).

The normalized image is calculated based on the mean and root mean square deviation of the original image, where M and VAR - the output values of the mean and the mean square deviation, are calculated by the formulas (9) (10):

$$M = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I(i,j), \quad (9)$$

$$VAR = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left(I(i,j) - M\right)^2. \quad (10)$$

2) Orientation image is calculated from the normalized image. It is defined as an image where $O(i,j)$ - the local orientation (angle of inclination) of the projection in the pixel with the coordinates (i,j) (11):

$$O(i,j)=\frac{1}{2}arctg\left(\frac{d_x^2(i,j)d_y^2(i,j)}{2d_x(i,j)d_y(i,j)}\right), \quad (11)$$

where $d_x(i,j)$ and $d_y(i,j)$ - gradients of the pixel with the (i,j) coordinates on the axes X and Y respectively.

3) From the normalized image, the frequency image F is calculated. Frequency image is an image where $F(i,j)$ - the local frequency of the protuberance, which is defined as the frequency of papillary lines of the image of the fingerprint. If due to the features of the papillary pattern, it is not possible to determine the pixel frequency, then its frequency is defined as the average value of the frequency of adjacent blocks.

Let I - the number of pixels between two adjacent vertices of the crests in the dimension block $W\times W$ with the center of which is a pixel with coordinates (i,j), then the frequency in this pixel will be (12):

$$F(i,j)=\frac{1}{I}., \quad (12)$$

4) Binaryization of the image. We will define a binary image R as an image $N\times N$ showing the (i,j) pixel category. The pixel may be a pixel of the hollow or pixel of the ridge.

$R(i,j)=1$, if $G(i,j)>R_0$, in all other cases $R(i,j)=0$ where $G(i,j)$ - the threshold of masking, R_0 - the intensity of the pixel of the normalized image.

5) The use of Gabor filters configured for the local orientation of the speeches, applies to the normalized input image (13):

$$G(x,y)=exp\left(-\frac{1}{2}\left(\frac{x_\varphi^2}{\sigma_x^2}+\frac{y_\varphi^2}{\sigma_y^2}\right)\right)cos\left(2\pi\theta x_\varphi\right), \quad (11)$$

where $x_\varphi=x\cos(\varphi)+y\sin(\varphi)$; $y_\varphi=-x\sin(\varphi)+y\cos(\varphi)$; φ - the orientation of the Gabor filter, θ - the frequency of the sinusoidal plane wave, σ_x^2 and σ_y^2 - the space constants of the Gaussian bypass along the axes x and y, respectively. These constants are established and adjusted on the basis of empirical data on the operation of the algorithm.

For the Gabor filter, you need to set up a Fourier transform, which gives the frequency information contained in the signal, that is, what the content of each frequency in the signal is. The integral is taken from $-\infty$ to $+\infty$ to the entire time axis. For the Fourier transform, it is equally true whether there is a certain frequency throughout the signal being studied, or it arose at a certain time, its contribution will still be the same.

Fourier transform is not suitable for the analysis of nonstationary signals, with one exception, when we are interested only in frequency information, and the time of existence of spectral components is not important. To correct these shortcomings Gabor's transformation can be used. Let (14):

$$g_a(t)=\frac{1}{2\sqrt{pa}}e^{\frac{-t^2}{4a}}, \quad (12)$$

where a is a fixed parameter. The function g_a is used as the time window.

Fig. 6 shows the Gabor function, which is a composition of the cosine and exponentials.

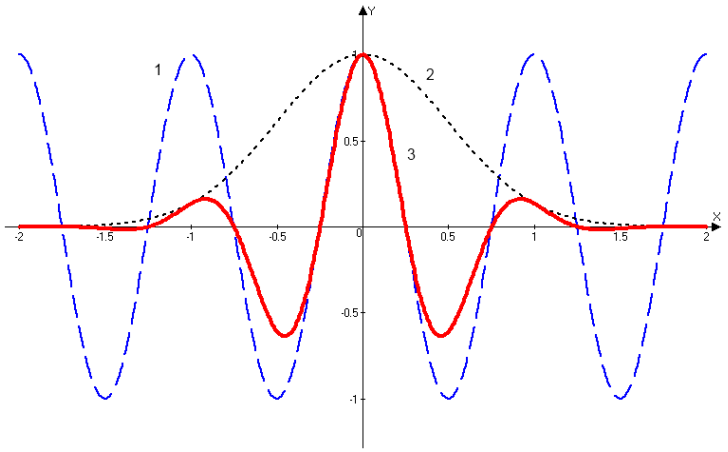


Fig. 6 - The function of the Gabor (3), which is a composition of the cosine function (1) and exponent (2)

Indeed, Gabor's transformation localizes the Fourier transform around a point $t = b$. To construct a one-dimensional Gabor filter, we use the formula (15):

$$G(x) = exp\left(-x^2 / 2\sigma^2\right) \times cos(2\pi\theta x), \quad (13)$$

where σ - the standard deviation of the Gaussian nucleus, which determines the amplitude of the function, θ is the frequency of oscillation, defined as $\theta = \frac{1}{T}$, where T - the period of function $cos(2\pi\theta x)$. The more is σ, the more gentle the form will accept the function.

5 Modeling and analysis of local features

Consider a heuristic approach based on the method of structural and statistical analysis. The main idea of the proposed model lies in the formation of the space of independent attributes when recognizing images given in the form of images. To describe each source image, methods are used to search the main components of the element (that is, the definition of structural features) and to calculate the statistical characteristics of this image of the fingerprints.

The proposed model of operators for identifying the characteristic features of fingerprint images when identifying a person includes the stage of determining the set of fingerprint features. At this stage, a set of characteristic features of fingerprints is formed, which are determined using both the structural method and the statistical method of determining the characteristic features of the images. In determining the structural features of fingerprints (for example, capillary lines of the imprint and special points of the prints), the structural method is used, and when calculating the various statistical characteristics of the image under consideration (for example, texture characteristics based on statistics of the first order), the statistical method. As a result of this stage we will get a set of features that characterize fingerprints. At the next stage there is a selection of subsets of strongly related features. At this stage, the system of "independent" subsets of attributes is determined. The following is a definition of representative features in each subset of characteristic strongly related features. At this stage, from each subset of such signs a single sign is determined and as a result of the implementation of this stage a set of representative features is distinguished. As a result of this stage we get a reduced space of signs, the dimension of which is much smaller. The final step is to identify the desired features. As a result of this stage, the best signs of fingerprints are formed.

Thus, a model of operators of the formation of the space of signs on the images of fingerprints is constructed. In the process of solving the practical problem it is determined that the stages of formation of subsets of "independent" features, namely, the question of determining the number of these subsets and a set of attributes on the image of fingerprints, as well as choosing a model of recognition, are important in solving the problem. Therefore, it is necessary to continue the study based

on the identified features. The developed model can be used in the compilation of various software complexes, focused on solving tasks of classification of objects, given in the form of images.

To solve the problem of identifying local features in this work it is proposed to apply a mathematical model of the salesman problem.

The mission of a salesman is important and difficult to solve. It represents the problem of finding the shortest Hamiltonian path in a complete finite graph with N vertices. In order to apply a mathematical device in solving a problem, it is necessary to present it as a mathematical model. The salesman's task can be presented as a graph using the vertices (minutias) and edges (the distance between the minutias). Let i, j - the vertices, and ribs (i, j) - the paths of communication between the points. In this case, for each of the edges you can match the criterion of the utility of the route $c_{ij} \geq 0$.

The Hamiltonian cycle can be called a route that involves passing through each vertex of the graph exactly once. In order to simplify the task and guarantee the existence of such a route, it is necessary that the model graph is fully connected. All known methods for finding the exact solution include searching for a space of decisions, which expands exponentially depending on N. The mission of a salesman can be solved with heuristic, search and precise algorithms.

Exact algorithms include the algorithm of full-fledging and the method of branches and boundaries.

The exhaustive search algorithm searches for $N!$ space solutions by scrolling through all the options. The result of the algorithm is the exact solution. The disadvantage of this algorithm is its temporal complexity - the search space grows exponentially, so when N is heuristic and search algorithms are not significantly less used.

Experimentally, the complexity of this algorithm was evaluated as $t = 0{,}0056 \cdot e^{(1{,}3789 \cdot N)}$ [14].

The advantages of this algorithm include the possibility of parallelization and the exact solution of the problem.

The method of branches and boundaries is the development of exhaustive search algorithm. Its essence is to add a test of the criterion that limits the functions and proceeds from the task at which, at a certain level, you can pause the construction of this branch of the permutation tree.

It retains all the positive properties of the exhaustive search algorithm, but nevertheless is not suitable for tasks where N is not very small. Experimentally, the complexity of this algorithm was evaluated as $t = 0{,}0745 \cdot e^{(0{,}8485 \cdot N)}$ [14].

In the case of use as a minimal initial solution, a solution obtained by the "greedy" method is used. The complexity of the algorithm will be $t = 0{,}3164 \cdot e^{(0{,}7469 \cdot N)}$ [14]. The advantages of this algorithm include the possibility of parallelization and the exact solution of the problem.

Heuristic algorithms include the method of incorporating a remote and BV-method.

The idea of incorporating a remote method is that the minutias, which are as far apart as possible, will never be adjacent to the chain. These two points will be the basis for further resolution. Then again there is a vertex that is as far removed as possible from the vertices already enclosed in the chain. There is a minimum sum of the lengths of the edges between the vertex found and the pair of adjacent vertices in the chain, which sets the place in the chain of the found vertex. This algorithm has a linear complexity, gives an approximate solution to a problem and can not be parallelized. His temporal complexity was appreciated as $t = 28{,}0600 + (-1{,}6069 \cdot N) + (0{,}0227 \cdot N^2)$ [14].

The BV method is based on an analysis of the existing reference route and its optimization. The decision can be divided into two stages: obtaining the original reference solution and optimizing the initial solution. The initial solution is the best of all decisions made on the basis of the "greedy" method. The second stage is to modernize the resulting initial reference route with the help of BV modifiers, which allow to identify non-optimal areas and convert them. This algorithm has a quadratic complexity, gives an approximate solution and can be parallelized at the 2nd stage [15]. His temporal complexity was appreciated as $t = -169{,}40 + (15{,}5786 \cdot N) + (-0{,}4104 \cdot N^2) + (0{,}0040 \cdot N^3)$ [14].

Search algorithms. The Genetic Algorithm and the Ant Colony System (ACS) algorithm are "leaders" among search algorithms [14].

The most optimal (result / time) among search algorithms is Genetic Algorithm. However, it also has its disadvantages associated with premature convergence (it is not always possible to find a way out of the local minimum). Experimentally its temporal complexity was estimated as $t = 683 + (-42{,}467 \cdot N) + (1{,}0696 \cdot N^2)$ [14].

Ant Colony System (ACS) is the development of the Ant System (AS) algorithm. Its main differences are:

- in the function of choosing a new point the balance between the use of accumulated knowledge and the study of new possible solutions is clearly set;
- at global renewal of pheromone (at the conclusion of each iteration) its addition occurs only to the arcs belonging to the global shortest path.

Heuristic algorithms are found much faster than the search algorithms. This is connected, as a rule, with the linear organization of the method itself and allows them to be used in those tasks, when computing time is a critical parameter. Exact methods are little suitable for solving problems of large size (unable to solve a task in a reasonable time), and search algorithms are a compromise between heuristic and precise methods [14].

6 Conclusion

Thus, the mathematical models and operations investigated allow us to formulate a list of unique biometric features that can be applied in user authentication systems. Structurally, the proposed model of the biometric fingerprint authentication system consists of the following steps.

In **the first stage**, it is necessary to generate normally distributed random variables. Since the normal distribution law is encountered in nature very often, so for it effective methods of modeling have been developed. In this paper, we propose using the method of taking the inverse function. This method is especially convenient to use in the case when the integral law of probability distribution is given analytically and a possible analytical take of the inverse function from it.

At the next, the **second stage**, it is necessary to process the data from the matrix formed at the previous stage. Since we are only interested in key points (they will correspond to "1" in the matrix), then at this stage the task of finding the shortest path will be solved, that is, a chain of points, sorted by minimum distances between the points, will be created.

In the **third stage**, the algorithm "Method of branches and boundaries" will be used to solve the salesman problem. Such an optimization algorithm is one of the effective polynomial algorithms for finding approximate solutions of the salesman problem, as well as solving similar problems in finding routes in graphs. The general idea of the method can be described in the example of finding the minimum of a function $f(x)$ on the set of valid variable x values. For the method of branches and boundaries, two procedures are required: branching and finding the marks (boundaries). The branching procedure consists in splitting the set of permissible variable x values into subsets of smaller sizes. The procedure can be applied recursively. The resulting subsets form a search tree or tree of branches and boundaries. The nodes of this tree are constructed subsets of values of a variable x. The procedure for finding estimates is to find the upper and lower bounds for solving the problem on a subset of admissible values. Discrete optimization methods, in particular branches and boundaries, allow finding optimal or approximate solutions for quite large tasks. The result of this stage is an integer expression, which is the optimal way in the graph, with vertices of which there are minutias.

On the last, fourth stage, the formation of a passive sequence occurs. To do this, as one of the options, the SHA-256 hexing algorithm will be applied to the integer expression. With regard to the use of hexing during the formation of a passive sequence, such a method requires additional research in the further development of work. Technical characteristics of the SHA-256 hex function: The length of the message digest is 256 bits, the length of the internal state is 256 (8×32) bits, the block length is 512 bits, the iteration cycles are 64.

The scheme of the computational algorithm of the developed software implementation of the biometric fingerprint authentication system is shown in Fig. 7.

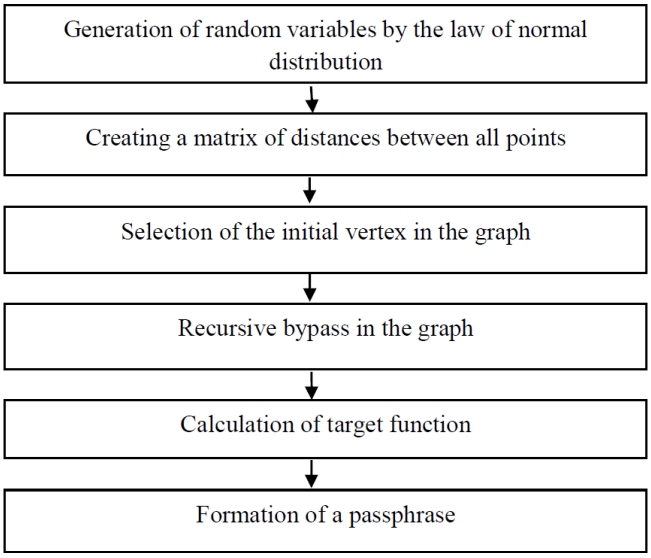


Fig. 7 - Scheme of computational algorithm

The developed software performs the transformation of fingerprint images, formed by the normal law of random variables, with subsequent transformation into a passive sequence. Consequently, it can be provided for modeling the fingerprint authentication system in order to investigate certain properties of it, to evaluate the probability indicators of efficiency (probabilities of the first and second kind errors), etc. These and other studies are a promising direction for our further scientific works.

References

[1] Дактилоскопия и генотипоскопия в судебной медицине. URL: http://www.forens-med.ru/book.php?id=105

[2] Криміналістичне дослідження слідів рук (дактилоскопія). URL: http://pidruchniki.com/2015060965282/pravo/ kriminalistichne_doslidzhennya_slidiv_ruk_daktiloskopiya

[3] Shvets V., Fesenko A. Basic biometric characteristics, modern systems and technologies of biometric authentication. *Ukrainian Scientific Journal of Information Security*. 2013. Vol. 19. P. 2 – 103.

[4] Рыканов А.С. Анализ методов распознания отпечатков пальца. *Системи обробки інформації*. 2010. Вип.6 (87). С. 164 – 171.

[5] Chernychenko I.V. Separate aspects of the formation of fingerprinting until 1900. *Bulletin of criminal justice*. 2005.№ 3. P. 136.

[6] Romanov V. O., Galelyuk I.B., Klochan P.S. Technologies of person's authentication by biometric characteristics. *Computer means, networks and systems*. 2010. № 9. P. 54 – 61.

[7] Зайцев В.Г., Степенко К.С. Спосіб контролю доступу на підставі розпізнавання відбитків пальців. URL: http://pmk.fpm.kpi.ua/arhive_2009/36-Stepenko.pdf

[8] Moroz A.O. Biometric Technologies. Methods of fingerprinting. *Mathematical Machines and Systems.* 2011. P. 3 – 64.

[9] Медична інформатика URL: http://intranet.tdmu.edu.ua/data/kafedra/internal/informatika/classes_stud/uk/med/biol/ptn.htm

[10] Вітлінський В.В. Моделювання економіки. Моделювання випадкових величин як системотвірна імітаційного процесу моделювання. URL: http://ecolib.com.ua/article.php?book=17&article=1540

[11] Вихман В.В., Якименко А.А. Биометрические системы контроля и управления доступом в задачах защиты информации: учебно-метод. пособие. Новосибирск: Изд-во НТГУ, 2016. 54 с.

[12] Руденко В.М. Математична статистика. URL: http://pidruchniki.com/18421120/statistika/normalniy_rozpodil

[13] Випадкові величини. Робоча навчальна програма дисципліни «Теорія ймовірностей та математична статистика» для підготовки бакалаврів за спеціальністю 6.091500 – Комп'ютерні системи та мережі. URL: http://elib.lutsk-ntu.com.ua/book/knit/vm/2011/11-47/page6.html

[14] Борознов В.О. Исследование решения задачи коммивояжера. *Вестник Астраханского государственного технического университета. Сер. Управление, вычислительная техника и информатика.* 2009. С. 147–151.

[15] Борознов В.О. Исследование эвристического метода решения задачи коммивояжера. *Электронный журнал "Исследовано в России»*. 2008. С.322 – 328. URL: http:// zhurnal.ape.relarn.ru/articles/2008/028.pdf

Рецензент: Сергій Толюпа, д.т.н., проф., Київський національний університет імені Т. Шевченка, м. Київ, Україна. E-mail: tolupa@i.ua

Надійшло: Лютий 2019.

Автори:

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

Катерина Будянська, студентка факультету комп'ютерних наук, Харківський національний університет імені В.Н. Каразіна, майдан Свободи 4, Харків, 61022, Україна.
E-mail: budyanskaya96@gmail.com

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

Михайло Багмут, аспірант, факультету комп'ютерних наук, Харківський національний університет ім. В.Н. Каразіна, майдан Свободи 6, Харків, 61022, Україна. E-mail: sapsanmiha@gmail.com

Математична модель біометричної системи автентифікації відбитків пальців.

Анотація. У роботі розглядаються математичні моделі біометричних образів відбитків пальців, а також основні обчислювальні процедури дактилоскопії. Досліджено основні етапи обробки дактилоскопічних портретів, засновані на виділенні локальних ознак, їх фільтрації і цифровій обробці. Розроблене програмне забезпечення реалізує перетворення образів відбитків пальців з подальшим формуванням на їх основі криптографічно потужної парольної послідовності. Це дозволяє моделювати систему дактилоскопічної автентифікації з метою дослідження її певних властивостей, оцінки імовірнісних показників ефективності (ймовірностей помилки першого і другого роду), тощо.

Ключові слова: відбитки пальців, біометричний образ, парольна автентифікація, біометрична система.

Рецензент: Сергей Толюпа, д.т.н., проф., Киевский национальный университет имени Т. Шевченко, г. Киев, Украина. E-mail: tolupa@i.ua

Поступила: Февраль 2019.

Авторы:

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

Екатерина Будянская, студентка факультета компьютерных наук, Харьковский национальный университет им. В. Н. Каразина, пл. Свободы 6, Харьков, 61022, Украина. E-mail: budyanskaya96@gmail.com

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

Михаил Багмут, аспирант факультета компьютерных наук, Харьковский национальный университет им. В. Н. Каразина, пл. Свободы 6, Харьков, 61022, Украина. E-mail: sapsanmiha@gmail.com

Математическая модель биометрической системы аутентификации по отпечаткам пальцев.

Аннотация. В работе рассматриваются математические модели биометрических образов отпечатков пальцев, а также основные вычислительные процедуры дактилоскопии. Исследованы основные этапы обработки дактилоскопических портретов, основанные на выделении локальных признаков, их фильтрации и цифровой обработке. Разработанное программное обеспечение реализует преобразование образов отпечатков пальцев с последующим формированием на их основе криптографически сильной парольной последовательности. Это позволяет моделировать систему дактилоскопической аутентификации с целью исследования определенных ее свойств, оценки вероятностных показателей эффективности (вероятностей ошибки первого и второго рода), и тому подобное.

Ключевые слова: отпечатки пальцев; биометрический образ; парольная аутентификация; биометрическая система.

УДК 004.4: 004.056.5 DOI: 10.26565/2519-2310-2019-1-02

ДЕФІНІЦІЙНІ ПРОБЛЕМИ ТЕРМІНОЛОГІЇ У СФЕРІ КІБЕРБЕЗПЕКИ І КІБЕРОБОРОНИ ТА ШЛЯХИ ЇХ ВИРІШЕННЯ

Сергій Вдовенко, Юрій Даник, Сергій Фараон

Національний університет оборони України імені Івана Черняховського, пр-т Повітрофлотський, 28, Київ, 03049, Україна
vsg64@ukr.net, zhvinau@ukr.net, faraon34@ukr.net

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

Надійшло в лютому 2019

Анотація: *На підставі аналізу термінології сфери кібербезпеки та кібероборони, національних інтересів України в кіберпросторі та з урахуванням досвіду провідних країн світу, у статті розглянуті концептуальні підходи щодо врегулювання нормативно-правового поля і термінологічних систем національного сектору кібербезпеки та кібероборони держави.*

Ключові слова: *кібератака; кібербезпека; кібервплив; кіберконфлікт; кіберпростір; кіберзахист; кіберзагроза; кіберзброя.*

1 Вступ

Результатом стрімкого науково-технічного прогресу у сфері інформаційних технологій стало значне посилення ролі складних автоматизованих систем управління, які застосовуються у багатьох галузях діяльності людини, зокрема у військовій сфері, в тому числі – для управління військовими операціями та зброєю. Процеси функціонування таких систем відбуваються у сформованому новому віртуальному просторі – кіберпросторі, який доповнив існуючи: сухопутний, морський, повітряний, космічний, та став сферою конфліктів і можливих бойових дій [1,2]. При цьому відбувається зміна традиційних форм і способів ведення протиборства. Більше того, майбутня війна може бути спровокована в кіберпросторі.

Відомо, що війна та військовий конфлікт не є кінцевою метою, а лише силовим інструментом, за умов неможливості досягнення визначених політичних та/чи економічних цілей іншими способами.

У 2010 році 1-й командувач Кіберкомандування Сполучених Штатів Америки (США) генерал К. Александер відмічав: «Домінування в кіберпросторі, на відміну від, наприклад повітряного не розглядалося при військовому плануванні. доки нові технології не надали таку змогу» [3]. Розвиток сучасних технологій надає змогу досягти перемоги навіть без безпосереднього зіткнення бойових компонентів. Перенесення збройного протиборства в інформаційно-інтелектуальну й інформаційно-технічну сфери суттєво підвищує роль і значення кібероборони.

Концептуальні проблемні питання щодо загроз національній безпеці, зокрема у сфері інформаційної безпеки, окремі організаційно-правові засади протидії кіберзлочинності та боротьби з кібертероризмом досліджували О.А. Баранов [4], В.Л. Бурячок [5,6,7,8], Ю.І. Грицюк [9], Р.В. Грищук [10], Ю.Г. Даник. [10], Д.В. Дубов [11,12,13], Р.В. Лук'янчук [14,15], В.В. Петров [16], В.П. Шеломенцев [17], М.Ю. Яцишин [18] та інші.

Деякі технічні та правові аспекти проблем захисту військової інфраструктури від деструктивних, в тому числі кібервпливів, розглядалися іноземними та вітчизняними фахівцями. [4, 9,19,20].

Соціальні аспекти кіберконфліктів досліджувалися, зокрема, О. Косенковим, якій відмітив, що засоби кібервпливу це перші в історії людства засоби боротьби, які реально існують та застосовуються, але без повного розуміння і контролю. За його думкою слід відрізняти

соціальні аспекти кіберконфліктів від інформаційної війни в кіберпротсторі, а кіберконфлікти слід вважати найважливішим компонентом війни [21].

Проблеми кібероборони, з точки зору воєнно-політичного та воєнно-стратегічного аналізу розглядаються здебільш іноземними фахівцями, публікуються в офіційних виданнях саммітів НАТО (*North Atlantic Treaty Organisation* - Організація Північноатлантичного договору), але не мають юридичної сили та є лише поглядами відповідних фахівців [22,23].

Всі поточні дослідження не розглядають цілісний підхід до проблем кібероборони.

Сучасна геополітика вимагає цілеспрямованої активної діяльності держав, щодо пошуку ефективної моделі оперативного управління кібербезпекою та підвищення ролі і значення державних інституцій щодо створення ефективної системи кібероборони, та реалізації заходів з її забезпечення. Особливо актуально це в умовах ведення гібридних війн, кризових ситуацій, надзвичайного або воєнного стану.

2 Основна частина

В Україні для досягнення необхідних оперативних спроможностей сектору безпеки і оборони між іншими, визначені завдання щодо: удосконалення систем інформаційної і кібербезпеки, систем захисту інформації та безпеки інформаційних ресурсів; забезпечення інформаційної і кібербезпеки; посилення спроможностей зміцнення інституціональних та технічних можливостей суб'єктів сектору безпеки та оборони для ефективної боротьби із кіберзагрозами воєнного характеру, кібершпигунством, кібертероризмом та кіберзлочинністю; поглиблення міжнародного співробітництва у цій сфері; формування підрозділів забезпечення кібербезпеки та кіберзахисту Збройних Сил України (ЗСУ); здійснення міжвідомчої координації та взаємодії з цих питань в інтересах забезпечення обороноздатності держави; створення необхідних матеріально-технічних ресурсів для забезпечення здатності протидіяти іноземним технічним розвідкам, інформаційним, кібернетичним атакам, спецопераціям противника; створення ефективних сучасних зразків кіберзброї; розвиток мережі реагування на комп'ютерні надзвичайні події (CERT) [24,25].

Стратегія кібербезпеки України [26] пріоритетними заходами у сфері забезпечення кібербезпеки сектору безпеки і оборони визначає:

- здійснення воєнно-політичних, військово-технічних та інших заходів для розширення можливостей Воєнної організації держави, сектору безпеки і оборони у кіберпросторі;
- створення, розвиток сил, засобів та інструментів можливої відповіді на агресію у кіберпросторі, яка може застосовуватись як засіб стримування військових конфліктів та загроз у кіберпросторі (*активний кіберзахист*);
- створення єдиного підрозділу із забезпечення кібербезпеки та кіберзахисту ЗСУ на стратегічному, оперативному та тактичному рівнях;
- розвиток підрозділів кібербезпеки та кіберзахисту ЗСУ;
- розвиток науково-виробничого потенціалу та системи підготовки спеціалістів кібербезпеки та кібероборони для потреб органів сектору безпеки і оборони України.

Разом з тим, довгостроковими та середньостроковими планами України [27,28] передбачається здійснити удосконалення та розвиток системи кібербезпеки та захисту інформації шляхом створення в Міністерстві оборони України (МОУ), інших складових сектору оборони підрозділів з кіберзахисту, протидії технічним розвідкам, впровадження заходів із захисту інформації відповідно до вимог нормативно-правових актів України та з урахуванням стандартів НАТО і ISO/IEC, причому у ЗСУ переважно за рахунок реформування Головного управління зв’язку та інформаційних систем Генерального штабу ЗСУ. Притому, ряд заходів передбачених Планом [28] суперечить нормативно-правовому полю держави. Це в деякій мірі звужує спектр визначених Указами Президента України [24-26] завдань та не сприяє створенню системи кібероборони.

Виходячи з цього, необхідність побудови ефективної кібероборони слід розглядати через призму ризиків та загроз, що притаманні світові та насамперед для України у першій чверті XXI століття.

В січні 2018 року в Сенаті США здійснено доповідь [29], в якій відзначається, що з 2014 року кіберпростір України використовується в якості театру кібердій та полігону для випробовування кіберзброї, а кібератаки, як головний інструмент гібридної війни, направлені на всі сектори суспільства та економіки, зокрема такі як медіа, фінанси, транспорт, політика, енергетика і військова справа. Визнається необхідність використання такої ситуації для розуміння тактики й практики дій, прогнозування типів майбутніх кіберударів та відпрацювання ефективних заходів захисту від них, з одночасним наданням допомоги Україні в побудові оборонних сил. Співпраця з Україною щодо протидії цим загрозам вважається критично важливим елементом створення кібероборони США [29].

При цьому відверто заявлено, що США та їх європейські союзники залишаються вразливими до кібератак, які здійснюються переважно з застосуванням гібридної асиметричної зброї та це є загрозою для всіх держав. За наявності загроз кібератак для урядових та неурядових структур міжнародне співтовариство не розробило правил, які могли б встановити норми керівництва кібероборoною у довгостроковій перспективі. Держави НАТО не визначили критерії класифікації кібератак в контексті можливості застосування для колективного захисту статті 5 Північно-Атлантичного договору.

Констатовано, що уряд США не має інституцій, здатних активно залучати сили й допомагати неурядовим організаціям щодо протидії загрозам кібератак, крім обговорення можливостей щодо здійснення киберзагроз на стратегічному рівні. Визначено, що Уряд (США), спільно з державами-членами НАТО, повинен переглянути масштаби наслідків кібератак, не тільки як кримінальних загроз, але й як загроз національній безпеці, має збільшити обмін інформацією між їх розвідувальними та правоохоронними органами й розробити офіційні керівні принципи стосовно того, яким саме чином Альянс буде розглядати такі напади, у контексті статті 5 Північно-Атлантичного договору [29]. Слід зазначити, що необхідність вжиття перерахованих заходів була декларована США в липні 2016 року у підсумковому документі Варшавського саміту НАТО [1].

В вересні 2018 року президентом США Д. Трампом підписана Національна кіберстратегія США (*National Cyber Strategy of the United States of America*), на підставі якої, а також на підставі Стратегії національної безпеки США, в цьому ж місяці анонсована Кіберстратегія МО США (*Department of Defense Cyber Strategy*), що замінює Кіберстратегію 2015 року. Ця Стратегія має обмеження доступу, відомі лише її загальні цілі та задачі. На відміну від попередніх Стратегій, остання ґрунтується на взаємопов'язаних завданнях: зі створення більш руйнівної кіберсили, здатної конкурувати з противником та стримувати його дії в кіберпросторі; з реформування системи управління Кіберсилами; з розширення співробітництва в межах альянсу та з партнерами; з підтримки та розвитку талантів у кіберсфері. Згідно доповіді командувача Кіберкомандування США генерал-лейтенанта П. Накасоне перед підкомітетом з кіберзахисту Сенату, Кіберстратегія МО США визначає наступні пріоритети: підготовка простору операцій; висока готовність; ресурсне забезпечення; підготовка; партнерство. Всі вищезазначені Стратегії у своїй структурі мають глосарій, якій містить скорочення, абревіатури, терміни та визначення [30,31]. Включення або не включення дефініцій до глосаріїв Стратегій є підставою внесення або виключення їх до/зі Словника МО США *(DOD Dictionary of Military and Associated Terms)* [32], якій є затвердженою термінологією для застосування всіма суб'єктами МО США та має юридичну силу стандарту.

В розвиток доповіді [29] у лютому 2018 року Палата представників Конгресу США схвалила проект «Закону про співпрацю з Україною з питань кібербезпеки», що спрямований на просування активнішої взаємодії між Україною і США у сфері кібербезпеки в умовах протидії загрозам у всесвітній мережі Інтернет.

Стратегія [26] передбачає гармонізацію нормативних документів України у сфері кібербезпеки відповідно до міжнародних стандартів і стандартів ЄС та НАТО. При цьому дуже важливим індикатором готовності систем кібербезпеки та кібероборони держав-партнерів є досягнення визначеного рівня їх інтероперабельності.

Аналіз існуючих Законів України і інших нормативно-правових актів України та провід-

них країн світу, зокрема США, свідчить про дефініційну, термінологічну та нормативно-правову невизначеність або/та розбіжність об'єктно-предметної області декількох десятків понять, що складають базовий термінологічний набір терміносистеми сфери кібербезпеки та кібероборони, зокрема таких як: *– кібербезпека; – кіберзахист; – кіберзброя; – кібероборона; – кіберпростір; – кібертероризм* тощо.

Семантичне навантаження дефініції визначається описом об'єкту, предмету, їх ознак та взаємозв'язків між ними. Гармонізація термінології сфери кібербезпеки та кібероборони вимагає однакового тлумачення суті об'єкту та предмету.

Крім того, недостатньо з'ясовано масштаб та наслідки можливого застосування кібертероризму, а також сил кібербезпеки (кібервійськ, кіберсил) провідних країн, що формуються та набувають оперативних спроможностей на протязі останніх років, їх функції, завдання, зміст діяльності, склад, порядок підготовки підрозділів, військових і цивільних фахівців у цій сфері [33]. У цьому контексті можна стверджувати, що аналіз проблем нормативно-правового, науково-технічного, організаційного і кадрового забезпечення розвитку кіберсил є актуальним для створення в Україні національної системи кібербезпеки та кібероборони.

Для досягнення зазначеної мети слід оперувати чітко визначеними поняттями та критеріями оцінки небезпеки (загрозами), Розглянемо деякі з них.

Кіберпростір. Поняття "кіберпростір" (*cyberspace*) вперше використано у 1984 р. американським письменником Уільямом Гібсоном ("Нейромант") для позначення всієї сукупності інформації, що міститься у комп'ютерних мережах. Доктрина інформаційних операцій ЗС США 2006 р. (*JP 3-13 Information Operations Doctrine*) це підтверджувала: "Кіберпростір – віртуальна обстановка, в якій цифрова інформація циркулює в комп'ютерних мережах".

Перше офіційне визначення кіберпростору було дано військовими експертами США в настанові КНШ 2006 року "Інформаційні операції": "Кіберпростір – сфера, в якій застосовуються різні радіоелектронні засоби (зв'язку, радіолокації, розвідки, навігації, автоматизації, управління і наведення), що використовують широкий діапазон електромагнітного спектра частот для прийому, передачі, обробки, зберігання, перетворення і обміну інформацією, і пов'язана з ними інформаційна інфраструктура ЗС США".

З розвитком цифрових технологій поняття в англомовній сфері та у деяких російських виданнях розширилось до позначення сукупності всіх електронних систем.

Так, згідно [34] *кіберпростір* – сукупність користувачів, мереж, пристроїв, програмного забезпечення, процесів, збереженої або транзитної інформації, додатків, послуг та систем, які можуть бути прямо чи опосередковано під'єднані до мереж. Словник [32] визначає кіберпростір, як глобальний домен в інформаційному середовищі, що складається з взаємозалежних мереж інфраструктури, інформаційних технологій і резидентних даних, включаючи Інтернет, телекомунікаційні мережі, комп'ютерні системи та вбудовані процесори та контролери.

Натомість, Законом України [35] визначено, що кіберпростір - це середовище (віртуальний простір), яке надає можливості для здійснення комунікацій та/або реалізації суспільних відносин, утворене в результаті функціонування сумісних (з'єднаних) комунікаційних систем та забезпечення електронних комунікацій з використанням мережі Інтернет та/або інших глобальних мереж передачі даних.

Тобто за кордоном кіберпростір розглядається як сфера діяльності складних технічних систем, а в Україні – складних соціотехнічних систем.

За поглядами окремих військових фахівців США домінування у кіберпросторі також виходить за рамки телекомунікаційних та інформаційних технологій і потребує переваги в усіх його складових: соціальній, технічній, телекомунікаційній, інформаційній, мережекомпьютерній тощо та по всьому електромагнітному спектру – "від постійного струму до денного світла, включаючи радіохвилі, інфрачервоне і рентгенівське випромінювання, спрямовану енергію, а також області, про які ще не почали навіть замислюватись, для забезпечення глобального командування і управління, глобального доступу і глобальної могутності"[36]. Тому, складовими кіберпростору слід вважати: інформаційний простір, комунікаційний простір, віртуальний комп'ютерно-мережний простір та соціотехнічний простір.

Розглядаючи сферу оборони (військовий аспект), визначимо, що кіберпростір – це єдиний простір сформований з інформаційного, комунікаційного, віртуального комп'ютерно-мережного та соціотехнічного просторів та об'єднаний системою зв'язків, в якому відбувається створення, зберігання, модифікація та передача інформації, управління об'єктами (системами) та зброєю, вплив на об'єкти (системи) протидіючої сторони, захист власних об'єктів (систем) в існуючих фізичних полях та середовищах.

Кібербезпека. Виходячи з дефініції "кібернетична безпека", єдиною і об'єднуючою ознакою в усі епохи розвитку людської цивілізації, яка однозначно характеризує явища та факти пов'язані з проблематикою її забезпечення, є безумовно ознака, яка визначає наявність систем та процесів управління. Виокремлення кібербезпеки в окремий вид безпеки сталося порівняно недавно. Вперше в світі в 1996 р. у військовій доктрині США "Concept Force XXI" на законодавчому рівні визнано необхідність захисту кіберпростору.

Так як проблема кібербезпеки носить глобальний характер, важливою є позиція міжнародних організацій. Так, Міжнародний союз електрозв'язку (ITU) [34,37] визначає що кібербезпека - це набір засобів, стратегії, принципи забезпечення безпеки, гарантії безпеки, керівні принципи, підходи до управління ризиками, дії, професійна підготовка, практичний досвід, страхування і технології, які можуть бути використані для захисту кіберсередовища, ресурсів організації та користувача. Загальні завдання безпеки у кіберсередовищі включають забезпечення ***доступності, цілісності, конфіденційності*** інформації. Глобальна програма [37] включає 5 стратегічних напрямів та 7 стратегічних цілей, що їх слід враховувати при створенні системи кібероборони, причому вимога щодо уніфікації глобального законодавства у сфері кібербезпеки розглядається як головна стратегічна ціль (див. Табл. 1).

За поглядами американських військових фахівців [38] до цього часу кібербезпека США розглядалася як комплекс заходів, спрямованих на захист комп'ютерів, електронних даних і мереж їх передачі від несанкціонованого доступу (конфіденційність), та інших дій, пов'язаних з маніпулюванням або крадіжкою, блокуванням (доступність), псуванням (спотворення), руйнуванням і знищенням (цілісність) умисного і випадкового характеру. Згідно Словника [32] виданого в січні 2019 року кібербезпека (безпека кіберпростору) - це дії, вжиті в захищеному кіберпросторі для запобігання несанкціонованому доступу, експлуатації або пошкодженню комп'ютерів, електронних систем зв'язку та інших інформаційних технологій, включаючи інформаційні технології платформи, а також інформацію, що міститься в ній, для забезпечення її доступності, цілісності, аутентифікації, конфіденційності і неспростовності. Зазначений приклад наведений з метою підтвердження зміни за останні три роки базового термінологічного апарату МО США в сфері кібербезпеки на 25-30%, що свідчить про гнучкість термінологічної системи сфери кібербезпеки США. Українське ж законодавство [39] комплекс цих заходів однозначно визначає як - ***технічний захист інформації*** (ТЗІ).

В Україні термін "кібербезпека" вперше використано у 2007 році [40], але лише в контексті необхідності "розробки та впровадження національних стандартів та технічних регламентів застосування інформаційно-комунікаційних технологій, гармонізованих з відповідними європейськими стандартами, у тому числі згідно з вимогами ратифікованої Верховною Радою України Конвенції про кіберзлочинність". З 2016 році кібербезпека України визначається як стан захищеності життєво важливих інтересів людини і громадянина, суспільства та держави в кіберпросторі [26]. Відповідно до Закону України [35], кібербезпека - захищеність життєво важливих інтересів людини і громадянина, суспільства та держави під час використання кіберпростору, за якої забезпечуються сталий розвиток інформаційного суспільства та цифрового комунікативного середовища, своєчасне виявлення, запобігання і нейтралізація реальних і потенційних загроз національній безпеці України у кіберпросторі.

Тобто, іноземні фахівці розглядають кібербезпеку як комплекс заходів та засобів захисту складних технічних систем, у той час як в Україні кібербезпека визначається як стан захищеності людини, громадянина, а також складних соціотехнічних систем.

На погляд авторів, основними критеріями ефективності заходів щодо забезпечення кібербезпеки повинні бути критерії, що базуються на оцінці якості функціонування саме складних

соціотехнічних систем. Оскільки, якщо реалізація кіберзагроз навіть і призводить до порушення роботи комп'ютерних систем, але це майже не позначається на якості функціонування відповідної соціотехнічної системи, то гострота проблеми забезпечення кібербезпеки різко знижується.

Таблиця 1 – Стратегічні напрямки і цілі міжнародної програми кібербезпеки ITU

Стратегічні цілі \ Стратегічні напрямки		Стратегічні напрямки				
		Правові заходи	Технічні та процедурні заходи	Організаційні структури	Створення потенціалу	Міжнародне співробітництво
Стратегічні цілі	Розробка стратегій формування уніфікованого глобального законодавства сумісного з діючими національними та регіональними нормами законодавства.	+			+	+
	Розробка глобальних стратегій утворення національних й регіональних організаційних структур та політики боротьби з кіберзлочинністю.		+	+	+	+
	Розробка глобальних стратегій вироблення критеріїв безпеки та вимог щодо санкціонування використання апаратних засобів, програмних засобів та систем.		+		+	+
	Розробка стратегій міжнародної координації діяльності щодо утворення глобальних структур спостереження, оповіщення, реагування на інциденти		+	+	+	+
	Розробка глобальних стратегій щодо утворення та затвердження універсальної системи цифрової ідентифікації, а також організаційних структур для забезпечення визнання цифрових посвідчень особи без урахування географічних кордонів.	+	+	+	+	+
	Розробка глобальної стратегії сприяння утворенню людського та інституційного потенціалу для збільшення знань та ноу-хау стосовно всіх цілей та напрямків.	+	+	+	+	+
	Підготовка пропозицій щодо засад глобальної стратегій участі всіх зацікавлених сторін задля міжнародного співробітництва, діалогу та координації діяльності стосовно всіх цілей та напрямків.	+	+	+	+	+

Кіберзагроза. Загрози - будь-які обставини або події, що виникають у зовнішньому середовищі та можуть призвести до небезпеки. Історичний досвід розвитку людської цивілізації показує, що невід'ємною ознакою існування останньої є ризики та потенційні загрози, які існують навіть тоді, коли про них нічого невідомо.

Кіберзагрози, носять глобальний характер. Межі кіберпростору не визначаються державними кордонами, або іншими географічними бар'єрами. Географічні, кліматичні, часові характеристики, місцезнаходження, державна (коаліційна) належність, форма власності об'єктів тощо, не є стримуючими факторами для здійснення кібервпливу чи кібератак. Кіберзагрози можуть бути реалізовані будь-де та в будь-який час та за незначний проміжок часу нанести величезні збитки. Потенційно вразливими до кіберзагроз є життєво важливі сектори економіки (енергетика, транспорт), критично важливі об'єкти інфраструктури, об'єкти критичної інформаційної інфраструктури, національна телекомунікаційна мережа, національні

електронні інформаційні ресурси, системи: банківсько-фінансова та охорони здоров'я, сфера оборони тощо. Правових і технічних заходів на національному та регіональному рівнях недостатньо для того, щоб подолати ці глобальні загрози.

Стандарт ITU та Європейського союзу ISO/IEC 27000 визначає загрозу (*threat*) як потенційну причину небажаного інциденту, що може призвести до збитків системі або/та організації [41]. Законодавство України [35] визначає кіберзагрози як наявні та потенційно можливі явища і чинники, що створюють небезпеку життєво важливим національним інтересам України у кіберпросторі, справляють негативний вплив на стан кібербезпеки держави, кібербезпеку та кіберзахист її об'єктів.

В Україні визнано [25], що у середньостроковій перспективі намагання реалізації іноземними державами, міжнародними злочинними угрупованнями кіберзагроз щодо автоматизованих систем державного та військового управління, об'єктів критичної інформаційної інфраструктури залишаються серед найбільш актуальних. Серед чинників, що вливають на результати протидії кіберзагрозам воєнного, кримінального, терористичного та іншого характеру відмічається недостатня ефективність діяльності суб'єктів сектору безпеки і оборони України у цій сфері. Крім того, кіберзагрози в оборонній сфері законодавчо або підзаконними актами в Україні не визначені.

Проблема оцінки стану кібербезпеки повинна передусім розглядатися в нерозривному зв'язку з оцінкою можливих чи завданих збитків соціальним або соціотехнічним системам відповідно до потенційних або/та реалізованих загроз. Найбільш небезпечні виклики та загрози національній безпеці України в сфері кібербезпеки визначені в [42].

Ряд системних та управлінських проблем слід розглянути окремо, як такі, що не забезпечують належний рівень (*індекс*) кібербезпеки:

- відсутність системності ведення кібердій, теорії застосування сил та засобів кібероборони, взаємодії різних відомств у сфері забезпечення кібероборони держави;
- відсутність в секторі оборони єдиного координуючого органу з питань забезпечення кібербезпеки та системи підготовки військового й цивільного персоналу;
- проблеми кадрового забезпечення відповідних структурних підрозділів та відтік за кордон кваліфікованих спеціалістів;
- зниження рівня наукового потенціалу, відсутність наукових шкіл, складнощі із методичним, науковим і технічним забезпеченням відтік за кордон кваліфікованих наукових кадрів.

Вирішення вище перерахованих проблем неможливе без гармонізації та унормування термінологічних систем сфер кібербезпеки та кібероборони. На фоні не вирішення таких проблем, потенційні кіберзагрози можуть реалізуватися в успішні кібератаки на складні соціотехнічні системи держави, які призведуть до виникнення критичних ситуацій (у тому числі техногенних аварій та катастроф. Зокрема, це можуть бути:

- порушення управління державою та її інституціями шляхом здійснення деструктивних впливів на соціум (населення та політиків і керівників різного рівня з метою усунення та дискредитація осіб, які приймають рішення, формування негативної громадської думки про дії влади, спонукання населення до деструктивних дій тощо);
- використання глобальних інформаційних мереж терористичними та екстремістськими організаціями з метою організації терористичних актів, а також вербування нових бойовиків;
- несанкціоноване втручання в комп'ютерні мережі та системи управління органів державного та військового управління, стратегічно важливих об'єктів критичної інфраструктури, національних підприємств, управління військами та зброєю з метою отримання доступу до службової, конфіденційної або комерційної інформації, її викрадення, спотворення чи знищення, або/та взяття таких систем під контроль чи виведення їх з ладу.

Терміни ***кібератака, кіберзахист, кіберрозвідка, кібертероризм, кібершпигунство*** – розглядаються, як це визначено в Законі [35]. Ряд дефініцій, які відсутні або некоректні в нормативно-правовому полі України можуть, на погляд авторів, бути визначними так:

*Кібероборона***.** Дефініція виразу кібероборона вимагає розуміння, що ключовим словом є оборона, а кібер - це зазначення простору, де відбуваються дії сил протиборчих сторін.

Класичне військове мистецтво розглядає оборону як вид воєнних (бойових) дій військ (сил), в основі яких є захисні дії ЗС, їх об'єднань, частин та підрозділів.

За масштабом оборона може бути стратегічною, оперативною і тактичною. За типом організації – вимушеною або навмисною. За показниками активності – активною або пасивною.

Досвід всіх воєн та військових конфліктів свідчить, що успішною може бути тільки активна оборона з елементами наступальних та інших видів дій військ (сил). Активність оборони характеризується масовою часткою таких дій. За класифікацією простору ведення воєнних дій – наземна оборона, оборона морського узбережжя та морських комунікацій, протиповітряна, протикосмічна тощо. Відповідно до організації збройних сил та ступеню інтеграції управління можуть розглядатися і інші види оборони.

Враховуючи визнання кіберпростору п'ятою сферою ведення воєнних (бойових) дій кібероборона розглядається як сукупність політичних, економічних, соціальних, військових, наукових, науково-технічних, інформаційних, правових, організаційних та інших заходів, які здійснюються в кіберпросторі та спрямовані на забезпечення захисту суверенітету та обороноздатності держави, запобігання виникненню збройного конфлікту та відсіч збройній агресії [1,2,35]. На погляд авторів, законодавчо визначена дефініція потребує корегування. Цілком зрозуміло, що політичні, економічні, соціальні, правові, організаційні заходи, які спрямовані на досягнення мети кібероборони, здійснюються не в кіберпросторі. Доцільно запропонувати наступне визначення: Кібероборона – сукупність політичних, економічних, соціальних, військових, наукових, науково-технічних, інформаційних, правових, організаційних та інших заходів, які здійснюються в Державі та кіберпросторі й спрямовані на забезпечення захисту її суверенітету та обороноздатності, запобігання виникненню збройного конфлікту та відсіч збройній агресії.

За висновками американської компанії *Communications and Electronics Association* у книзі «The First Information War», виданої у 1992 році, застосування комп'ютерних мереж та високих технологій для управління логістикою в операції "Буря в пустелі" забезпечувало майже миттєвий обмін інформацією та змінило війну.

Кібервійна (війна у кіберпросторі) – складне суспільно-політичне явище, що відбивається в протиборстві непримиренних сторін в кіберпросторі з використанням кіберзброї, засобів кіберрозвідки, захисту або впливу для завдання матеріальних втрат (збитків) супротивнику і мінімізації особистих втрат (збитків) в економічній, військовій, політичній та ідеологічних сферах.

Кібервплив – сукупність реалізованих підрозділами кібероборони за єдиним замислом та під єдиним керівництвом одночасних або послідовних взаємопов'язаних за метою і завданнями кібератак та/або кіберударів, спрямованих на визначені елементи кіберпростору противника з метою порушення їх функціонування (стану), або - на елементи управління через кіберпростір з метою порушення процесів управління. Його складові: програмно-комп'ютерний вплив; фізичний вплив на органи і системи управління; радіоелектронне подавлення (ураження); інформаційно-психологічний вплив тощо.

Кіберзброя – сукупність технічних, програмних та інших засобів, призначених для здійснення деструктивних впливів на визначені елементи кіберпростору противника з метою виведення їх з ладу, або - на елементи управління через кіберпростір з метою порушення процесів управління.

Кібероперація – сукупність спланованих і реалізованих визначеними силами та засобами за єдиним замислом та під єдиним керівництвом одночасних або послідовних взаємопов'язаних за метою і завданнями кібератак та/або кіберударів, спрямованих на визначені елементи кіберпростору противника, або - на елементи управління через кіберпростір, з одночасним захистом власного кіберпростору від таких дій з боку противника.

Кіберудар – форма воєнних (спеціальних) дій підрозділів кібероборони спрямованих на реалізацію деструктивних (руйнівних) впливів на визначені елементи кіберпростору противника з метою їх блокування, знищення елементів інфраструктури, об'єктів, техніки та озброєння, руйнації циркулюючих в них даних і інформації, як в реальному так і в розподіленому

масштабі часу. За масштабами може бути стратегічного або оперативного рівня. За ступенем концентрації зусиль – поодинокий, масований. За вибором цілі – цільовий, груповий.

З 2011 року Положення Міжнародної стратегії кібербезпеки [38] розглядають кібератаки на критично важливу інфраструктуру як "акт війни", що підпадає під статтю 5 Північноатлантичного договору та дає юридичні підстави для удару у відповідь будь-якими засобами відповідно до ситуації, включно традиційними військовими. Словник [32] визначає кібероборону (оборону кіберпростору) як заходи, вжиті в захищеному кіберпросторі для подолання конкретних загроз, які порушили або загрожують порушенням заходів безпеки в кіберпросторі, і включають заходи щодо виявлення, визначення типів та характеристик загроз (включно шкідливе програмне забезпечення або несанкціоновані дії користувачів), протидії їм та мінімізації їх наслідків, приведення систем до безпечної конфігурації.

З урахуванням широкого застосування сучасних інформаційних технологій у секторі безпеки і оборони, створення єдиної автоматизованої системи управління ЗСУ оборона держави стає більш уразливою до кіберзагроз [2].

Згідно Законів [43,44] оборона України, захист її суверенітету, територіальної цілісності і недоторканності, охорона повітряного простору та підводного простору держави покладаються на ЗС України. Однак жодним Законом України кіберпростір не визначений, як середовище ведення оборонних дій для забезпечення захисту суверенітету держави. Натомість Закон України [35] встановлює, що національна система кібербезпеки включає в тому числі й оборонні заходи, а також визначає МО України та Генеральному штабу ЗС України завдання щодо підготовки держави до відбиття воєнної агресії у кіберпросторі (кібероборони); військової співпраці з НАТО та іншими суб'єктами оборонної сфери щодо забезпечення безпеки кіберпростору та спільного захисту від кіберзагроз, забезпечення кіберзахисту критичної інформаційної інфраструктури в умовах надзвичайного і воєнного стану.

Стратегія [26] визначає МО України, Генеральному штабу ЗС України завдання щодо здійснення заходів з підготовки держави до відбиття воєнної агресії у кіберпросторі (кібероборони) та кіберзахисту власної інформаційної інфраструктури. Закон України (зі змінами) [45] визначає обов'язковість здійснення заходів з кібероборони (активного кіберзахисту) для захисту суверенітету держави та забезпечення її обороноздатності, запобігання збройному конфлікту та відсічі збройній агресії, у тому числі - проведення спеціальних операцій (розвідувальних, інформаційно-психологічних тощо) у кіберпросторі при підготовці до захисту та захисту України в разі збройної агресії.

А ні законами України, а ні іншими нормативно-правовими актами не визначено перелік вичерпних заходів щодо підготовки до відбиття та відбиття воєнної агресії у кіберпросторі. Тому, для досягнення мети та безпосереднього виконання заходів у сфері забезпечення кібербезпеки сектору безпеки і оборони, визначених в [24, 25, 26, 32, 35, 45] необхідно стандартизувати та гармонізувати в нормативно-правовому полі України дефініції термінологічних систем сфери кібербезпеки та кібероборони.

Операція формування значення для терміну, дефініція (лат. *definitio* - визначення), є важливим засобом описів та міркувань у наукових теоріях та галузях знань, чим виконує важливу функцію у науково-освітній та практичній діяльності [46].

В термінології, як розділі лексикології, аксіомою є, що визначення будь якого терміну (дефінієндума) та зміст і значення визначаючого поняття (дефінієнса) мають бути тотожними, вичерпувати один одного і мати один і той же зміст (денотат). До науково-технічних термінів висуваються додаткові вимоги: системність, вмотивованість, однозначність, точність, відсутність синонімів [47].

З певних історичних, воєнно-наукових, зовнішньополітичних та інших причин в термінологічній системі галузі кібербезпеки та кібероборони України, склалося протиріччя, що вимагають відповідного наукового розв'язання. Воно полягає в недотриманні в термінографії сфери кібербезпеки принципів однозначності, точності та відсутності синонімів. А саме, у терміносистемі сфери кібербезпеки одночасно існує й паралельно застосовується низка дефініцій, в яких одному дефінієндуму (Dfd) ставиться у відповідність декілька дефінієнсів (Dfn),

або навпаки, один дефінієнс розкриває значення різних дефінієндумів. Термінологічна сфера кібероборони в Україні ще не сформована, тим не менш, процесу її формування притаманні ті ж самі помилки. Ускладнення цього протиріччя в площині практичного застосування термінологічного апарату сфери кібербезпеки та кібероборони відбувається за рахунок невідповідності термінологічних систем сфер кібербезпеки міжнародного співтовариства, зокрема ЄС та НАТО й України.

Вирішення протиріччя полягає у формуванні за правилами науково-технічної лексикографії множини семантичних аналітичних та синтетичних визначень термінологічної системи сфери кібербезпеки та кібероборони. Автори вважають за можливе організацію та виконання даної роботи здійснити за наступним алгоритмом:

1. Вибір термінів та їх дефініцій з множини ключового набору термінів термінологічних систем сфер кібербезпеки та кібероборони

2. Збір максимальної кількості вживаних конкретних дефініцій кожного терміну терміносистеми, включно з іноземних офіційних джерел міжнародних організації та країн-партнерів

3. Аналіз кожної дефініції:

- на системність – то б то належність терміна до певної термінологічної системи; за результатом – виключення зайвих термінів;
- на відсутність синонімів; за результатом – виключення зайвих термінів, що в межах одній терміносистеми, забезпечує запобігання взаємному непорозумінню фахівців;
- на однозначність, тобто на тотожність тільки одного наукового або технічного терміну Dfd та відповідного йому поняття Dfn; за результатом – виключення зайвих термінів;
- на точність – при чому слід з’ясувати чому виникло занадто широке значення змісту (надлишковість) або занадто вузьке визначення;
- на вмотивованість, то б то спроможність передати змістовне навантаження без додаткового застосування термінологічного словника.

4. Декомпозиція обраних для подальшої роботи термінів.

5. Композиція однозначних нових дефініцій термінів. При чому, дефініція кожного нового словосполучення (складного терміну) має містити дефініційні ознаки кожного слова складного терміну, які мають формувати дефініцію складного терміну. При цьому складний термін має формувати нові властивості притаманні тільки йому.

6. Аналіз нової дефініції терміну на вмотивованість, точність, однозначність, відсутність синонімів, системність.

7. Порівняльний аналіз на точність та вмотивованість синтезованої дефініції терміну з аналогом міжнародної терміносистеми в даній сфері

8. Формування пропозицій щодо включення термінів до фахових термінологічних словників.

9. Формування пропозицій щодо гармонізації нормативних документів України у сфері кібербезпеки та кібероборони відповідно до міжнародних стандартів і стандартів ЄС та НАТО.

3. Висновки

У роботі проведений аналіз термінології сфери кібербезпеки і кібероборони та розглянуті концептуальні підходи щодо врегулювання нормативно-правового поля та термінологічних систем національного сектору кібербезпеки, та кібероборони держави.

Запропоновано алгоритм формування множини семантичних аналітичних і синтетичних визначень термінологічної системи сфери кібербезпеки та кібероборони, що враховує правила науково-технічної лексикографії.

Сформульовані рекомендації щодо усунення визначених невідповідностей шляхом нормативно-правового й дефініційного врегулювання понятійного апарату у сфері кібербезпеки та кібероборони.

Визначені напрями подальших досліджень, що пов'язані з формуванням остаточного переліку функцій та завдань суб'єктів кібероборони, систем управління, їх взаємозв'язків, критеріїв (індикаторів) загроз у сфері кібероборони держави.

Посилання

[1] Warsaw Summit Communiqué. Issued by the Heads of State and Government participating in the meeting of the North Atlantic Council in Warsaw 8-9 July 2016. URL: https://www.nato.int/cps/en/natohq/official_texts_133169.htm

[2] Стратегія національної безпеки України, затверджена Указом Президента України від 26.05.2015 № 287/2015. URL: https://zakon.rada.gov.ua/laws/show/287/2015

[3] Keith A. Statement for the Record, Commander, US Cyber Command: House Armed Services Committee Statement. (Washington, DC. 23 September 2010). URL: https://www.govinfo.gov/content/pkg/CHRG-111hhrg62397/pdf/CHRG-111hhrg62397.pdf

[4] Баранов О.А. Про тлумачення та визначення поняття "кібербезпека". *Правова інформатика.* 2014. № 2(42). С. 54 – 62.

[5] Бурячок В. Л. Основи формування державної системи кібернетичної безпеки: монографія. Київ: НАУ, 2013. 432 с.

[6] Бурячок В.Л. Кібернетична безпека – головний фактор сталого розвитку сучасного інформаційного суспільства. *Сучасна спеціальна техніка* : зб. наук. праць. 2011. № 3 (26). С. 104 – 114.

[7] Інформаційна на кіберберзпека: соціотехнічний аспект / Бурячок В. Л., Толубко В. Б., Хорошко В. О., Толюпа С. В. Київ: ДУТ, 2015. 288 с.

[8] Бурячок В. Л., Гулак Г. М., Дорошко В. О. Завдання, форми та способи ведення воєн у кібернетичному просторі. *Наука і оборона*. 2011. № 3. С. 35 – 42.

[9] Грицюк Ю.І. Кіберінтервенція та кібербезпека України: проблеми та перспективи їх подолання. *Науковий вісник НЛТУ України*. 2016. Вип. 26.8. URL: http://nltu.edu.ua/nv/Archive/2016/26_8/52.pdf

[10] Грищук Р.В., Даник Ю.Г. Основи кібернетичної безпеки: монографія. Житомир: ЖНАЕУ, 2016. 636 с.

[11] Дубов Д.В., Ожеван М.А. Кібербезпека: світові тенденції та виклики для України. Київ: Вид-во НІСД, 2011. 30 с.

[12] Дубов Д.В. Стратегічні аспекти кібербезпеки України. *Стратегічні пріоритети*: наук.-аналіт. щокварт. збірник Нац. ін-т стратег. дослідж. 2013. № 4 (29). С. 119–126.

[13] Дубов Д. В. Кіберпростір як новий вимір геополітичного суперництва: монографія. Київ: НІСД, 2014. 328 с. URL: http://www.niss.gov.ua/content/articles/files/Dubov_mon-89e8e.pdf

[14] Лук'янчук Р.В. Державна політика у сфері забезпечення кібернетичної безпеки в умовах проведення антитерористичної операції. *Вісник НАДУ*: зб. наук. праць. 2015. Вип. 3. С. 110 – 116.

[15] Лук'янчук Р. В. Деякі питання реформування системи державного управління у сфері забезпечення кібернетичної безпеки: сучасний погляд. *Вісник НАДУ*: зб. наук. праць. 2013. Вип. 2. С. 81 – 92.

[16] Петров В.В. Щодо формування національної системи кібербезпеки України. *Стратегічні пріоритети*: наук.-аналіт. щокварт. збірник Нац. ін-т стратег. дослідж. 2013. № 4 (29). С. 127–130.

[17] Шеломенцев В.П. Правове забезпечення системи кібернетичної безпеки України та основні напрями її удосконалення. *Боротьба з організованою злочинністю і корупцією (теорія і практика)*: зб. наук. праць. 2012. № 1(27). С. 312 – 320.

[18] Яцишин М. Ю. Міжнародно-правова протидія кібервійнам. *Збірник праць Національного авіаційного університету*. 2015. № 1. С. 67 – 71.

[19] Вдовенко С.Г., Даник Ю.Г. Концептуальні напрями комплексного вирішення проблеми захисту інформації в системі скритого управління Збройних сил. *Сучасні інформаційні технології у сфері безпеки та оборони.* 2017. № 2(29). С. 98 – 106.

[20] Вдовенко С.Г., Даник Ю.Г. Концептуальні напрямки комплексного вирішення проблеми захисту від несанкціонованого доступу в складних системах спеціального призначення. URL: http://epsi.vntu.edu.ua/uploads/2017/61-f1s0m4m1jhvk8ix2vjd81nyxuzb1i3q9.pdf

[21] Kosenkov A. Cyber Conflicts as a New Global Threat file. URL: futureinternet-08-00045.pdf

[22] Andress J., Winterfeld S., Rogers R. Cyber warfare: Techniques, tactics and tools for security practitioners . Amsterdam : Syngress/Elsevier, 2011. 289 p.

[23] The Tallinn Manual on the International Law Applicable to Cyber Warfare 2.0. Tallinn 2016. URL: http://csef.ru/media/articles/3990/3990.pdf

[24] Стратегія національної безпеки України, затверджена Указом Президента України від 26.05.2015 № 287/2015. *Урядовий кур'єр*. 2015. № 95. URL: http://zakon.rada.gov.ua/287/2015

[25] Концепція розвитку сектору безпеки і оборони України, затверджена Указом Президента України від 14.03.2016 №92/2016. URL: https://zakon.rada.gov.ua/laws/show/92/2016

[26] Стратегія кібербезпеки України, затверджена Указом Президента України від 15.03.2016 № 96. *Офіційний вісник України*. 2016. № 23.

[27] Стратегічний оборонний бюлетень України, введений в дію Указом Президента України від 06.06.2016 № 240/2016 URL: https://www.president.gov.ua/documents/2402016-20137

[28] План дій щодо впровадження оборонної реформи у 2016 - 2020 роках (дорожня карта оборонної реформи), затверджений Міністром оборони України 15.08.2016. URL: http://www.mil.gov.ua/diyalnist/reformi-ta-planuvannya-u-sferi-oboroni/22082016-04.html; http://www.mil.gov.ua/content/tenders/Plan_2208.pdf

[29] Putin's asymmetric assault on democracy in Russia and Europe: implications for U.S. National security a minority staff report prepared for the use of the committee on foreign relations United States Senate one hundred fifteenth congress second session January 10, 2018 . URL: http://www.gpoaccess.gov/congress/index.html

[30] DOD. Joint Publication 3-12, Cyberspace Operations, 8 June 2018. URL: https://fas.org/irp/doddir/dod/jp3_12.pdf

[31] Statement by lieutenant general Paul M. Nakasone Commander, United States Army Cyber Command before the Subcommittee on Cybersecurity Committee on Armed Services United States Senate second session, 115th congress May 23, 2017. URL: https://www.armed-services.senate.gov/imo/media/doc/Nakasone_05-23-17.pdf

[32] DOD Dictionary of Military and Associated Terms. As of January 2019. URL: https://www.jcs.mil/Portals/36/Documents/ Doctrine/pubs/dictionary.pdf

[33] Мазулевський О.В., Вдовенко С.Г. Огляд підходів дій кібернетичних сил держав світу. *Перспективи розвитку озброєння і техніки сухопутних військ*: збірник тез доповідей Міжнародної науково-технічної конференції. Львів: НАСВ, 2018.С.220. URL: http://repositsc.nuczu.edu.ua/bitstream/123456789/7088/1/17-18-05-2018_zb_tez_dop.pdf

[34] Рекомендация МСЭ-Т X.1205. Обзор кибербезопасности. Женева: МСЕ, 2010. С. 55. URL: www.itu.int/ITU-T/recommendations/rec.aspx?rec=9136&lang=ru

[35] Закон України «Про основні засади забезпечення кібербезпеки України» № 2163-VIII від 5 жовтня 2017 року. URL: https://zakon.rada.gov.ua/laws/show/2163-19

[36] Secretary of State Hillary Rodham Clinton On the Release of President Obama Administration's International Strategy for Cyberspace. May 16, 2011. URL: http://www.state. gov/secretary/rm/2011/05/163523.htm

[37] ITU Global Cybersecurity Agenda (GCA) A Framework for International Cooperation in Cybersecurity. URL: https://www.intgovforum.org/Substantive_2nd_IGF/ITU_GCA_E.pdf

[38] International Strategy For Cyberspace: Prosperity, Security and Openness in a Networked World. Washington DC: The White House, May 2011. URL: http://www.whitehouse.gov/sites/default/files/rss_viewer/internationalstrategy_ cyberspace.pdf.

[39] Про захист інформації в інформаційно-телекомунікаційних системах: Закон України. URL: http://zakon1.rada.gov.ua/cgi-bin/laws/main.cgi

[40] Стратегія національної безпеки України, затверджена Указом Президента України від 12.02.2007 № 105/2007 в редакції Указу Президента України від 8.06.2012 року № 389/2012 . URL: https://zakon.rada.gov.ua/laws/show/105/2007

[41] Рекомендації міжнародного союзу електрозв'язку. МСЕ-Т.Сер.Х.1208. Мережі передачі даних, взаємозв'язок відкритих мереж та безпека. Безпека кіберпростору – кібербезпека. 2014 р. ISO/IEC 27000. URL: https://www.itu.int/itu-t/recommendations/rec.aspx?rec=11950&lang=ru

[42] Кібербезпека: українські реалії. URL: http://timeua.info/post/oborona-i-bezopasnost/k-berbezpeka-ukra-ns-k--real--07454.html

[43] Конституція України. URL: http://zakon0.rada.gov.ua/laws/show/254

[44] Про Збройні Сили України: Закон України від 6 грудня 1991 року N 1934-XII (зі змінами). URL: http://zakon3.rada.gov.ua/laws/show/1934-12

[45] Про оборону України: Закон України станом на 01.07.2018 р., затверджений ВР України від 06.12.1991, № 1932-XII . URL: http://zakon4.rada.gov.ua/laws/show/1932-12

[46] Новейший философский словарь. URL:https://www.google.com/search?q=chrome.69i57.9536j0j8&sourceid=chrome&ie=UTF-8

[47] Васенко Л.А., Дубічинський В.В., Кримець О.М. Фахова українська мова: Навч. посібник. Київ: Центр навчальної літератури, 2008. 272 с. URL: http://uchebniks.com/book/277-faxova-ukrayinska-mova-navchalnij-posibnik-vasenko-la/23-vimogi-do-terminiv.html

Reviewer: Alexandr Kuznetsov, Doctor of Technical Sciences, Prof., Academician of the Academy of Sciences of Applied Radio Electronics, V. N. Karazin Kharkiv National University, 6 Svobody Sq., Kharkiv, 61022, Ukraine.
E-mail: kuznetsov@karazin.ua

Received: February 2019.

Authors:

Serhii Vdovenko, Colonel, Associate Professor of the Department of Communication and Automated Control Systems of the Institute of Information Technologies of the National University of Defense named after Ivan Chernyakhovsky, Kyiv, Ukraine.
E-mail: vsg64@ukr.net

Yury Danik, Major general, Doctor of Sciences (Eng.), Full Prof., Head of the Institute of Information Technologies of the National University of Defense named after Ivan Chernyakhovsky, Kyiv, Ukraine.
E-mail: zhvinau@ukr.net

Serhii Faraon, Colonel, Adjunct of the Scientific Department of the Organization of Training and Attestation of Scientific and Pedagogical Staff of the Scientific and Methodological Center for the Organization of Scientific and Scientific-Technical Activity of the National University of Defense of Ukraine named after Ivan Chernyakhovsky, Kyiv, Ukraine.
E-mail: faraon34@ukr.net

Definitive problems of the Terms of the Sphere of Cyber security and Cyber Defense and the Ways of their solution.

Abstract. Based on cybersecurity terminology analysis and cyber defense, national interests of Ukraine in cyberspace and taking into account the experience of leading countries of the world, the article discusses conceptual approaches to resolving the regulatory and definitive field in the state cyberdefense sector.

Keywords: Cyberattack; Cybersecurity; Cyberimpact; Cyberconflict; Cyberdefense; Cyberspace; Cyberthreat; Cyberweapons.

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

Поступила: Февраль 2019.

Авторы:

Сергей Вдовенко, полковник, доцент кафедры связи и автоматизированных систем управления Института информационных технологий Национального университета обороны Украины имени Ивана Черняховского, Киев, Украина.
E-mail: vsg64@ukr.net

Юрий Даник, генерал-майор, доктор технических наук, профессор, начальник института информационных технологий Национального университета обороны Украины имени Ивана Черняховского, Киев, Украина.
E-mail: zhvinau@ukr.net

Сергей Фараон, полковник, адъюнкт научного отдела организации подготовки и аттестации научно-педагогичных кадров научно-методического центра организации научной и научно-технической деятельности Национального университета обороны Украины имени Ивана Черняховского, Киев, Украина.
E-mail: faraon34@ukr.net.

Дефинитивные проблемы терминологии в сфере кибербезопасности и кибероборонь и пути их решения.

Аннотация. На основании анализа терминологии сферы кибербезопасности и кибероборонь, национальных интересов Украины в киберпространстве и с учетом опыта ведущих стран мира, в статье рассмотрены концептуальные подходы по урегулированию нормативно-правового и дефинитивного поля сектора кибероборонь государства.

Ключевые слова: кибератака; кибербезопасность, кибервлияние; киберконфликт; киберпространство; киберзащита; киберугроза; кибероружие.

УДК 004.031.43: 004.056 DOI: 10.26565/2519-2310-2019-1-03

ОБЗОР ПРОТОКОЛОВ КОНСЕНСУСА ПРИМЕНЯЕМЫХ В ТЕХНОЛОГИЯХ БЛОКЧЕЙН

Диана Ковальчук, Татьяна Ивко, Татьяна Кузнецова, Алексей Нарежний

Харьковский национальный университет имени В.Н. Каразина, пл. Свободы, 4, Харьков, 61022, Украина
dianakovalhyk@ukr.net, t.ivko@outlook.com, kuznetsova.tatiana17@gmail.com, o.nariezhnii@karazin.ua

Рецензент: Александр Потий, д.т.н., проф., Харьковский национальный университет имени В. Н. Каразина, пл. Свободы, 4, Харьков, 61022, Украина.
potav@ua.fm

Поступила: Март 2019

Аннотация. *Рассмотрены категории популярных проколов консенсуса: Proof of Work и его гибриды, Proof of Stake (включает LPoS, Proof of Importance) и гибриды, DAGs и его разновидности. В статье описаны их алгоритмы, характеристики, особенности, а также недостатки и преимущества. Уязвимости и атаки которым подвержены. Также приведен список использования каждого протокола в криптовалютах и других системах.*

Ключевые слова*: консенсус; блокчейн; протокол консенсуса блокчейн; децентрализованные системы; распределенный реестр.*

1 Введение

В 2009 году был сгенерирован первый блок и первые 50 биткойнов, а 1-я транзакция (перевод) произошла 12 января 2009 года. Появление биткоина положило начало технологии блокчейн [1]. В общем случае блокчейн – это цепочка блоков, каждый из которых обладает меткой времени, ссылкой на предыдущий блок и хранится на разных компьютерах.

Принцип работы блокчейна довольно прост. Его можно представить как некую «учетную книгу», которая есть у каждого участника события и которая постоянно обновляется. По сути, в эту книгу можно вписать любое событие – от финансовых операций с теми или иными криптовалютами и вплоть до результатов голосования на выборах или каких-либо идентификационных данных.

Интересная особенность блокчейна заключается в том, что страницы этой условной книги одновременно хранятся у всех пользователей сети, при этом постоянно обновляются и ссылаются на старые (предыдущие) страницы. И если кто-либо попытается скомпрометировать такую систему, несанкционированно удалив или изменив какую-либо запись, то система сразу же обратиться к десяткам тысяч других своих версий этой книги и обнаружит несоответствие в структуре блоков.

Консенсус является соглашением, которое удовлетворяет каждую из вовлеченных сторон. В контексте криптографии консенсус является процедурой принятия решения. Его цель – обеспечить всех участников сети возможностью согласования своего текущего состояния после добавления новой информации, блока данных или пакета транзакций. Иными словами, консенсус-протокол гарантирует, что сформированная цепь верна и подтверждает честность (легитимность) ее участников. Это важная структура для предотвращения ситуации, когда кто-то один контролирует всю систему, и она гарантирует то, что все участники соблюдают правила сети.

Протокол – это набор правил. Протоколы помогают:

- обеспечить стабильные условия для осуществления транзакций в сети;
- устранить возможность двойной траты;
- удостовериться, что все участники соблюдают предусмотренные правила.

Роль консенсусных алгоритмов заключается в обеспечении требуемого уровня надежности сети, построенной на серии узлов (*устройств, соединённых с другими устройствами, как часть компьютерной сети*). Консенсусные алгоритмы должны быть достаточно разви-

тыми, чтобы успешно предсказывать любые возможные сбои коммуникации внутри сети. Алгоритм автоматически спрогнозирует, что некоторые процессы и системы будут недоступны, и что в результате этого некоторые коммуникации будут потеряны. Чтобы противостоять этому, консенсусный алгоритм должен быть отказоустойчивым и работать для достижения заранее определенного консенсуса или одобрения, по крайней мере, от большинства узлов.

Блокчейн-системы, могут обладать только двумя из трёх возможных свойств: - децентрализация; - масштабируемость; - безопасность.

Каждый согласованный алгоритм имеет свой собственный сценарий применения, а выбор того, какой конкретно консенсус использовать для реализации блокчейна, зависит от типа сети и данных.

Чтобы транзакция была действительной в большинстве криптовалютных сетей, эта транзакция должна собрать определенное количество подтверждений (часто равных включению в блок цепочки блоков) из сети. Например, процесс получения 10 подтверждений означает просмотр конкретной транзакции в одном и 9 последовательных блоках.

2 Разновидности построения блокчейн реестра

2.1 Direct Acyclic Graph Tangle (DAG)

Это согласованный алгоритм DAG, используемый IOTA. Для того чтобы отправить транзакцию IOTA, необходимо проверить две предыдущие транзакции, которые вы получили. Консенсус «два к одному» с оплатой за продвижение усиливает достоверность транзакций, чем больше транзакций добавляется в направленный ациклический граф (tangle). Поскольку консенсус устанавливается транзакциями, теоретически, если кто-то может генерировать 1/3 транзакций, то он может убедить остальную часть сети, что их недействительные транзакции действительны. До тех пор, пока объем транзакции не станет достаточным, чтобы создание 1/3 тома стало невозможным, IOTA выполняет своего рода «двойную проверку» всех транзакций сети на централизованном узле - «Координаторе». Фактически, «Координатор» работает как стабилизатор (*предотвращает изменения параметров под действием дестабилизирующих факторов*) системы и будет удален, как только ациклический граф станет достаточно большим. В каждый момент времени существуют одна или несколько завершающих транзакций *(неподтвержденные транзакции - tips*), которые замыкают весь направленный граф существующих транзакций. При этом, разработчики утверждают, что при низких нагрузках на сеть количество замыкающих транзакций будет мало, а при высокой частоте появления новых транзакций число завершающих вершин будет возрастать [2].

Схематически структура реестра блокчейн на основе DAG представлена на рис. 1.

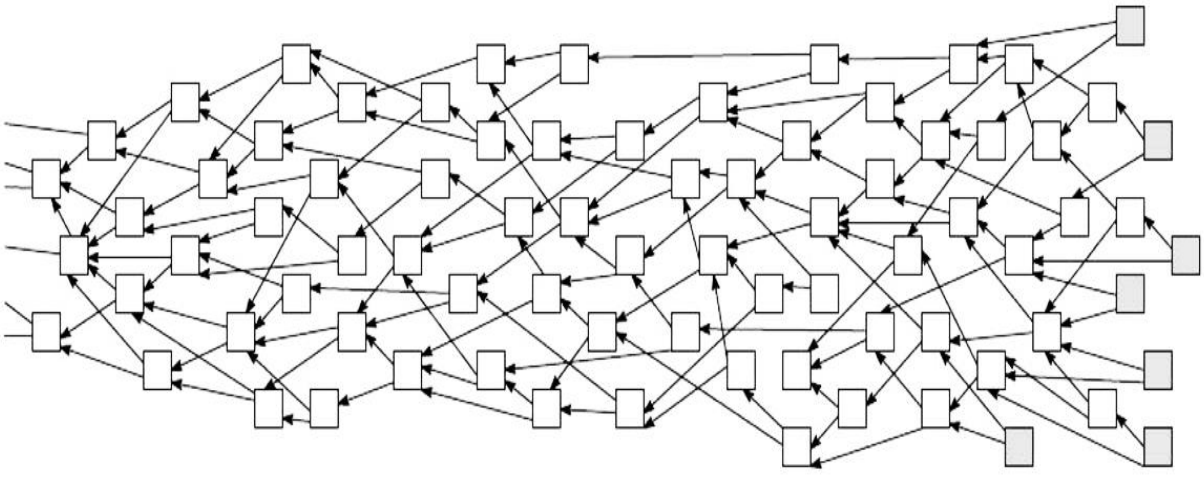


Рис. 1 – Структура реестра DAG

Используется в:
- IOTA [3].

Плюсы:
- низкие комиссии за транзакции;
- чрезвычайно малые транзакции;
- масштабируемость;
- облегченный.

Минусы:
- нет умных контрактов.
- уязвим (*требуется 34% от общей мощности хеширования*).

2.2 HashGraph

Принцип: узлы связываются случайным образом с использованием протокола «gossip about gossip» и соглашаются на консенсус после определенного раунда коммуникации.

Производительность: - очень высокая.

Среда DLT (Distributed ledger technology): - приватный блокчейн с разрешениями.

DLT - это технология хранения информации, ключевыми особенностями которой является: совместное использование и синхронизация цифровых данных согласно алгоритму консенсуса; географическое распределение равнозначных копий в разных точках по всему миру: отсутствие центрального администратора.

Завершенность: - зависит от раунда.

Ключевым отличием Hashgraph является протокол «*Gossip About Gossip*», в соответствии с которым узел получает набор транзакций с меткой времени, о которых «знает» другой узел. Для работы такого алгоритма все участники в сети должны быть известными. В результате синхронизации каждый узел хранит всю информацию и историю получения этой информации всеми узлами сети. Т.о., как только узел «видит» в своей истории, что данное сообщение уже было получено и проверено большинством, то нет сомнений, что оно действительно.

Используется в:
- Hedera Hashgraph [4].

Плюсы:
- быстрые транзакции (порядка 250 тыс. транзакций в сек).

Минусы:
- нет умных контрактов;
- не устойчив к атакам типа Sybil [5];
- не исследовано, как поведет себя в крупных масштабах использования.

2.3 Block-lattice - Directed Acyclic Graphs (DAGs)

Блочная решетка (*Block-lattice*) - это структура, в которой каждый пользователь (или адрес) получает свою собственную цепочку, в которую могут писать только они, но у каждого есть копия всех цепочек. Блочная решетка преобразует общий реестр (как в биткойн) в множество не используемых совместно асинхронных регистров, которые ускоряют время транзакций.

Блокчейн состоит из упорядоченных блоков (Рис. 2), которые содержат заголовки (*Header*) и транзакции (*Transaction*). Заголовок каждого блока, помимо других метаданных, содержит ссылку на своего предшественника в форме хеша предшественника. Начальное состояние жестко закодировано в первом блоке, называемом блоком генезиса. В отличие от других блоков, у блока генезиса нет предшественника.

В отличие от блоков, структура DAG хранит транзакции в узлах, где каждый узел содержит одну транзакцию (Рис. 3). В криптовалюте Nano каждая учетная запись связана со своей собственной цепочкой счетов в структуре, называемой блочной решеткой, эквивалентной истории транзакций.

Каждому аккаунту предоставляется цепочка аккаунтов. Цепочка учетных записей может рассматриваться как выделенная цепочка блоков, только для одной учетной записи. Узлы добавляются к цепочке счетов, а каждый узел представляет одну транзакцию в цепочке счетов. Аналогично блоку генезиса в блокчейне, DAG содержит транзакцию генезиса. Генезис транзакции определяет начальное состояние. В Nano вместо одной транзакции, которая передает значение, для полной передачи значения необходимы две транзакции.

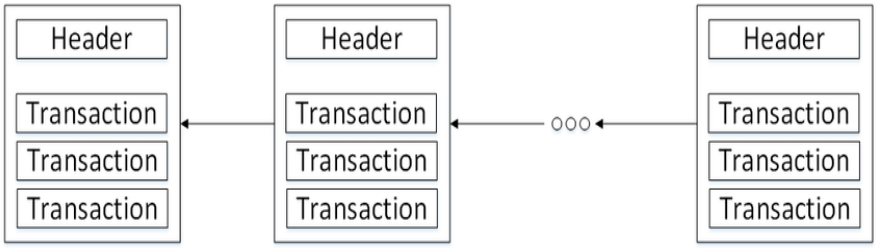


Рис. 2 - Структура блоков в Блокчейн

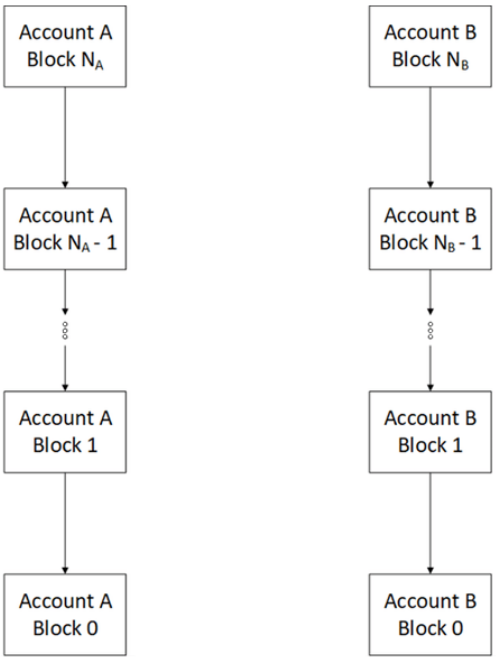
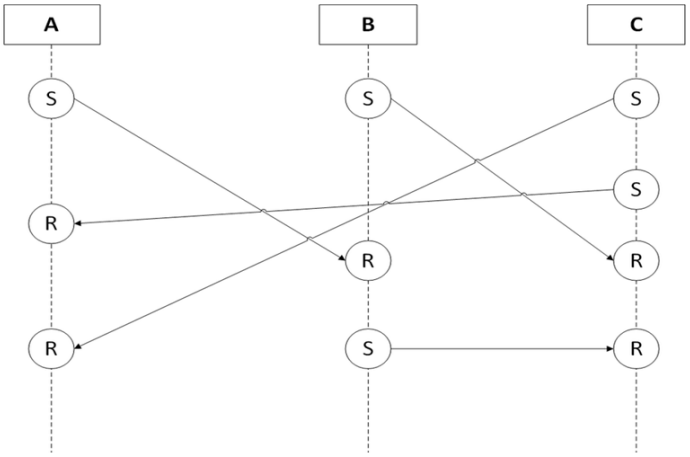


Рис. 3 - Структура блочной решетки в DAG



S - транзакция отправки;
R - транзакция приема.

Рис. 4 - Генерация транзакции отправки

Отправитель генерирует транзакцию отправки, в то время как получатель генерирует совпадающую транзакцию приема (Рис. 4). Когда выдается транзакция отправки, средства списываются с баланса счета отправителя, ожидая, пока получатель получит соответствующую транзакцию приема. В этом состоянии транзакции считаются неурегулированными. Когда транзакция получения сгенерирована, транзакция рассчитывается. Недостатком такого подхода является то, что узел должен быть подключен к сети для получения транзакции.

Обработка транзакций в блочной решетке.

Каждая транзакция разбивается на блоки отправки в цепочке отправителя и, соответственно, блок получения в цепочке получателя. Транзакция отправки вычитает средства из баланса отправителя, в то время как транзакция получения добавляет средства к балансу получающего счета. Если владелец аккаунта ведет себя подозрительно, то остальная часть

сети проголосует против недействительного блока, и он будет отклонен.

Каждая цепочка аккаунтов обновляется только владельцем аккаунта вследствие того, что каждый блок в цепочке должен быть подписан закрытым ключом учетных записей. Остальная сеть узлов по-прежнему будет подтверждать, что каждый блок является действительным (нет дубля расходов и пользователи не увеличивают свой баланс больше предусмотренного).

Nano использует Proof of Work (PoW), чтобы избежать спамеров, поскольку в сети не взимается плата за транзакции. Для работы с каждым блоком необходимо небольшое количество времени, примерно 5 секунд для генерации и 1 *мкс* для его проверки. Это заставляет злоумышленников выделять значительную вычислительную мощность для реализации атаки, в то время как всем остальным (легальным пользователям) требуется лишь небольшое количество вычислительных ресурсов. Также возможно удаление этих спам-транзакций, что ограничивает объем хранилища, которое можно использовать при атаках этого типа.

Атаки:

- *«Атака с потерей денег»*: злоумышленники увеличивают количество цепочек, которые должен отслеживать узел, отправляя незначительные суммы в широкий ряд пустых «кошельков».
- *«Двойная трата злоумышленника»*. Обе версии двойных расходов должны быть подписаны закрытым ключом пользователя. Следует определить учетные записи, которые отвечают за спам-атаки, а затем внести их в соответствующий «черный» список на определенный период времени.

Используется: - в Nano [6].

Плюсы:

- меньшие требования к хранению за счет сокращения базы данных, поскольку блокчейн каждого пользователя отслеживает баланс своего счета, а не суммы транзакций;
- блокчейн пользователя обновляется асинхронно с остальной частью блочной решетки;
- меньшее время транзакции (т.к. вся сеть не обрабатывает каждую транзакцию);
- нет комиссии за транзакции в сети.

Минусы:

- нет финансовых стимулов для запуска полного узла;
- Нано является дефляционной валютой (постоянно растет в цене);
- нет «умных» контрактов.

3 Алгоритмы консенсуса основанные на Proof of Work

3.1 Proof of Work (PoW)

Принцип: - трудно найти решение, но легко проверить результат.
Производительность: - низкая.
Среда DLT: - публичный блокчейн..
Завершенность: вероятностная.

PoW первоначально была создана как средство борьбы со спамом. Так, например, майнеры используют PoW для проверки транзакций, но его главная цель - блокировка потенциальных кибератак или подозрительных действий в сети. Суть данного алгоритма сводиться к двум основным пунктам:

- необходимости выполнения определенной достаточно сложной и длительной задачи;
- возможности быстро и легко проверить результат.

PoW задачи изначально не предназначены для их решения человеком. Решение выполняется компьютером, однако требует больших вычислительных мощностей. При этом важно то, что проверка полученного решения всегда требует гораздо меньше ресурсов и времени.

Майнеры

В криптовалютных сетях «майнеры» - это специальные узлы, которые выполняют вычисление PoW для набора транзакций, плюс хэш предыдущего блока для генерации следующего блока в цепочке блоков. Очевидно, что поскольку блок содержит хэш предыдущего блока, то

изменение «исторического» блока потребует регенерации всех последующих блоков. Таким образом, восстановление всех хэшей потребует большого объема вычислений и дополнительного расхода энергии (т.е. дополнительных финансовых расходов). Процесс использования аппаратных ресурсов компьютера для выполнения вычислений с целью подтверждения транзакций и обеспечения безопасности сети называется майнингом. Майнеры вознаграждаются за выполнение этих функций новыми монетами (*вновь создаваемыми биткойнами*).

Узлы

Узлы всегда считают, что самая длинная цепочка является правильной и продолжают работать над ее расширением. Если два узла одновременно транслируют разные версии одного блока, то некоторые узлы могут получить любой из них в первую очередь. В этом случае они работают над блоком, который был получен первым, но сохраняют и другую ветвь, на случай, если она станет длиннее. При обнаружении следующего доказательства работы, связь будет разорвана, т.к. одна ветвь станет длиннее, а узлы, которые «работали» в другой ветке, переключатся на более длинную ветвь.

Алгоритм

1. Транзакции связываются в виде блоков.
2. Майнеры проверяют транзакции внутри блоков на их подлинность.
3. Майнеры решают задачу, известную как «проблема доказательства работы».
4. Формируется награда первому, кто решит «проблему …».
5. Подлинные транзакции записываются в общедоступную цепочку блоков.

Атаки

Угроза централизации вычислительных мощностей, известная как атака 51%, считается одной из ключевых уязвимостей для алгоритма консенсуса PoW. Это происходит, когда у атакующей стороны, в роли которой может выступать сравнительно небольшое количество майнеров, находится «контрольный пакет» хэшрейта – вычислительной мощности сети.

Причиной данной уязвимости является тот факт, что майнеры могут одновременно предлагать сети верные хэши – решения, которые позволяют им подтверждать целостность данных и добавлять в сеть новые блоки. В этом случае в блокчейне происходит «разветвление». Алгоритм консенсуса PoW считает, что остальные майнеры признают верной ту ветвь, которая имеет наибольшее количество блоков, и проголосуют за ее включение в блокчейн. Таким образом, если майнер или совокупность майнеров контролируют больше половины хэшрейта, то у него/их появляется возможность добавлять свои ветви и тем самым манипулировать двусторонними операциями и не подтверждать новые транзакции.

Эта атака может привести к тому, что недобросовестные майнеры могут отзывать уже совершенные финансовые транзакции, что называется двойной тратой (*double-spending*). При этом атакующая сторона не может менять информацию в уже добавленных блоках и генерировать новые криптовалюты.

Плюсы:

- самый известный и самый безопасный;
- комиссия за транзакцию не обязательна;
- легко проверяемые решения;
- трудоемкость поиска решения;
- трудоемкость поиска решения может быть точно количественно оценена.

Минусы:

- низкая производительность;
- PoW характеризуется большим потреблением вычислительных мощностей, что само по себе снижает стимул;
- PoW уязвим для серьезных уязвимостей (например, атака 51 %);
- уменьшение награды за блок;
- доказательство работы ограничивает входные данные структурой алгоритма майнинга блокчейнов. В случае «Биткойна» это должен быть одноразовый номер, а в случае

«Эфириума» входными данными может являться случайное целое число, одноразовый номер и начальный хеш блока.

3.2 Hybrid Proof of Work (HPoW)

HPoW по-прежнему использует PoW, но модифицирует его, и как следствие, создает целую криптовалютную сеть, которая может работать на простых в настройке и недорогих компьютерах или облачных сервисах. HPoW устраняет стимул для майнеров, потому что награда за майнинг низкая. На самом деле, майнинговые фермы потеряли бы деньги, если бы попытались майнить в Lynx [7], где Lynx (LYNX) – криптовалюта или цифровой актив, а это значит, что они предоставят Lynx людям, которые хотят решить проблему устойчивости. Это забирает контроль у майнинговых ферм и отдает его в руки отдельных людей (индивидуальных майнеров), которые хотят работать на Lynx.

HPoW поддерживает обслуживание сети, стимулируя возможности тех, кто хочет использовать Lynx. С каждым подключаемым майнером сеть становится все более защищенной за счет снижения рисков, связанных с централизацией криптовалютной сети. Эта безопасность достигается, в том числе, за счет избыточности: т.е. чем больше отдельных узлов в сети, тем более устойчивой она становится. При этом если происходит сбой отдельного узла или майнера, или же если выходит из строя даже весь регион узлов (например, из-за перебоев в подаче электроэнергии или воздействия техногенных факторов), то сеть остается по-прежнему работоспособной и защищена, вследствие существования множества других ресурсов для майнинга.

Три бизнес-правила HPoW:

1. Один майнер не может выиграть блок чаще, чем раз в 30 минут.
2. Баланс адресов майнера должен быть больше или равен требуемому минимальному количеству для Lynx, чтобы выиграть блок.
3. При случайном выборе, самые «быстрые» майнеры не всегда гарантированно получают награду за блок.

Минусы:

- добыча убыточна.

3.3 Proof of Meaningful Work (PoMW).

Текущая реализация, использующая только искусственные вычислительные задачи (хеширование), слишком расточительна для рационального использования и плохо масштабируется. PoMW первоначально реализует хорошую идею, однако при ее реализации, используются значительные вычислительные ресурсы.

Основной минус**:** - атака более 50 % от общей вычислительной мощности.

3.4 Proof-of-Work-Time (PoWT).

PoWT – это относительно новый подход к формированию консенсуса, путем введения переменного времени блокировки, которое масштабируется с мощностью майнинга (*где блокчейн ускоряется с увеличением мощности*). Его использование лучше масштабирует блокчейн, увеличивает скорость транзакции и позволяет автоматически настраивать более прибыльный майнинг. Время блокировки зависит от сложности (max около 6,2 мин, min порядка 15 сек). Награды зависят от времени блокирования.

3.5 Proof of Edit Distance.

Алгоритм «Редактирование расстояний» - это класс алгоритмов, которые оценивают, насколько близко две строки находятся друг к другу. Например, расстояние редактирования для «ETH» (*Ethereum*) и «ETC» (*Ethereum Classic*) равно «0,8222», где две одинаковые строки будут иметь значение «1». Известно много алгоритмов, также в пространстве сходства строк, включая расстояние Левенштейна, расстояние Смита-Уотермана Гото и расстояние Рэтклифа-Обершелпа [8].

Использование Proof of Edit для создания новых блоков.

Майнеры конкурируют, чтобы найти строку, которая при хешировании в процессе нормализации превышает порог минимального расстояния. Эта строка может являться хэшем промежуточной цепочки блоков и заголовком хеша для следующего блока.

Пусть минимальное пороговое расстояние - «*t*», строка для поиска - «*B*», а редактированная функция расстояния – «ED», такая, что для каждого заголовка блокчейн удовлетворяет хэшу «*h*»:

$$ED\ (H\ (h),H\ (B))\ <t\ .$$

Или в случае объединения 2-х блокчейнов:

$$ED\ (H\ (h1),H\ (B))\ <t\ \&\&\ ED\ (H\ (h2),H\ (B))\ <t\ \ =\text{верно.}$$

Чтобы найти новый блок в промежуточной цепочке, майнер должен перебирать случайный набор символов (хэшируя строки), пока не найдется хэш, превышающий пороговое значение для всех блоков.

Плюсы:

- доказательством редактирования расстояния является независимый алгоритм майнинга;
- любой хеш или строковая структура могут быть предоставлены в качестве входных данных. Это означает, что пока блокчейн имеет уникальный хеш, его можно легко добавить в задачу «Доказательство редактирования».

Используется в Block Collider [9].

4 Алгоритмы консенсуса основанные на Proof of Stake

4.1 Proof of stake (PoS)

Принцип: - сеть доверяет валидатору, который предоставляет свои ресурсы в залог за возможность создавать блоки: - чем больше доля, тем выше вероятность того, что сеть разрешит создание блока.

Производительность: - высокая.

Среда DLT: - публичный/приватный блокчейн.

Завершенность: - вероятностная.

PoS работает с использованием алгоритма, который выбирает участников с самыми высокими ставками в качестве валидаторов, предполагая, что они заинтересованы в обработке транзакции. При этом у кого больше всего монет в обращении, больше всего и теряют, поэтому они готовы работать в интересах сети. Количество монет, которое сеть может потребовать изменяется, так же, как и сложность в PoW (рассматривалось выше).

В PoS блоки создаются не майнерами, выполняющими работу, а майнерами, которые ставят свои токены, чтобы «делать ставки» на то, какие блоки являются действительными. В случае разветвления, майнеры тратят свои жетоны на голосование, т.е какое ветвление поддерживать. Т.о. предполагая, что большинство пользователей голосуют за «правильную» вилку, валидаторы, проголосовавшие за неправильный вариант, «потеряют свою долю» в правильной вилке.

Общим аргументом против *proof-of-stake* является т.н. проблема «Ничто на кону» (*Nothing at Stake*). Она заключается в том, что валидаторы практически не требуют вычислительной мощности для поддержки разветвления (в отличие от PoW) и могут голосовать за обе стороны каждого ветвления. Т.о. форки в PoS могут быть гораздо более распространенными [9], чем в PoW, а это, как полагают некоторые специалисты, может снизить доверие к валюте.

Coin age (возраст монеты).

Чтобы различать пользователей, которые только что получили свои монеты, и пользователей, которые держали свои монеты в течение определенного периода времени, в алгоритмах доказательства ставки используется концепция возраста монет (*Coin age*).

Возраст монет используется, как в расчетах веса ставки, так и вознаграждения за ставку. Вознаграждение за ставку устанавливается APR монеты [10]. Данная характеристика имеет

постоянный интерес для всех кошельков со ставками, независимо от размера ввода или разумного времени простоя.

Чем дольше пользователь держит монеты, тем вероятнее возможность выиграть право создать блок сетевой цепочки блоков и получить вознаграждение.

Технология стимулирования пользователей.

Для поддержки сетевой активности и привлечения большего количества пользователей, награда за создание блока увеличивается, если в сети присутствует много пользователей.

Наказание пользователя за отсутствие активности (оффлайн).

Большинство, если не весь алгоритм PoS, наказывают держателей монет, которые остаются в автономном режиме в течение длительных периодов времени. В противном случае они могли бы получить контроль, получив более 50% прав голоса из-за размера своих владений.

Псевдо-анонимные пользователи.

Валидаторы в сети PoS - это анонимные пользователи, которые идентифицируются только по адресу своего кошелька. Это не дает никакой дополнительной ответственности по отношению к PoW плохим игрокам, которые могут накопить значительную выгоду в сети.

Безопасность.

Модель безопасности является сугубо экономической, основанной на том предположении «теории игр», что стоимость приобретения токенов, необходимых для перехода в статус производителя блоков, больше той стоимости, которую злоумышленник готов внести. Это обстоятельство связывает безопасность сети со значением ее токенов, т.е.: чем выше значение токена, тем более защищенной становится сеть.

Атака "Ничего на кону".

Без системы экономических штрафов для злоумышленников сеть может подвергнуться атакам «ничто на кону», когда заинтересованные лица мотивированы в проверке всех предлагаемых вилок для максимизации прибыли.

Алгоритм.

1. Валидаторы в качестве ставки блокируют часть своих монет.

2. Валидаторы инициируют проверку блоков. В случае обнаружения блока, который, по их мнению, может быть добавлен в цепочку, его подтверждают, делая на него ставку.

3. При добавлении блока в цепочку, валидатор получает вознаграждение, пропорциональное его ставке.

Плюсы:

- рентабельность: скорость, энергия, оборудование;
- чем большее число пользователей используют сеть и имеют монеты, тем безопаснее сеть;
- по отношению к другим более децентрализованный.

Минусы:

- характерно «экономическое неравенство», т.е. богатые становятся еще богаче;
- злоумышленники могут рассчитать вероятность получения награды, чтобы создать блок цепочки блоков, основываясь на анализе, у кого сколько монет;
- начальные цели для реализации Ethereum Casper - всего 100 TPS *(Transactions per Second)*;
- подвержен атаке «Ничего на кону».

Используется в:

- Tezos [11];
- - Decred [12];
- - Ethereum;
- - Peercoin;
- - Ada;
- - EOS.IO [13];
- Gridcoin;
- Nxt [14];

- Waves [15];
- BlackCoin;
- Qtum;
- In Future with Casper in Ethereum.

4.2 Гибридный PBFT/Aurand.

Polkadot использует гибридный механизм консенсуса PBFT/Aurand, который имеет две степени завершенности. Этот механизм допускает переходы состояний с относительно низкой задержкой при одновременном демпфировании некоторых видов атак.

PBFT обеспечивает полную завершенность, а Aurand обеспечивает быструю промежуточную завершенность. В каждой последовательности блоков есть контрольная точка PBFT, которая гарантирует завершенность, в то время как Aurand устанавливает промежуточные состояния завершенности.

С помощью Aurand один случайно выбранный валидатор может предложить блок, который «может быть отменен». Они ограничиваются, если предполагается, что блоки недействительны. Для достижения консенсуса требуется соотношение не менее PBFT > 2/3 подписанных валидаторов (для согласования блока).

Завершенность PBFT происходит ориентировочно каждые 30 сек, в то время как блоки Aurand предлагаются каждые 4-5 сек.

Используется в Polkadot.

4.3 Delegated proof-of-stake (DPoS).

Принцип: - участники делегируют производство новых блоков небольшому, фиксированному числу избранных валидаторов, чем обеспечивается высокая конкуренция и выгодность.

Производительность: высокая.

Среда DLT: - публичный/приватный блокчейн.

Завершенность: - вероятностная.

DPoS – это алгоритм достижения консенсуса в децентрализованой среде, который является альтернативой консенсусам PoW (*Bitcoin proof-of-work*) и PoS (*Peercoin или NXT proof-of-stake*).

Основной принцип работы DPoS - это разделение голосующих и валидирующих участников. В итоге, участники сети, имеющие право голоса в системе (держатели монет) не являются при этом валидаторами транзакций. Таким образом, одно подмножество участников выбирает другое подмножество, которое в свою очередь, будет формировать блоки. При этом возраст монеты не имеет значения.

Условия, в которых работает данный алгоритм, отличаются от условий, в которых работают PoW и PoS. При этом валидаторам необходимо раскрыть свою личность и заявить о готовности поддержкивать работу полноценного узла сети, своевременно выполнять верификацию транзакций и формировать новые блоки. DPoS сильно отличается от PoS: - в данном случае владельцы токенов не голосуют за достоверность самих блоков, а голосуют за то, чтобы избрать валидаторов для проверки от их имени.

При использовании консенсуса на основе модифицированного PoS каждый пользователь, по желанию, может выставить свою кандидатуру на пост верифицирующей рабочей станции (проверяющего узла - валидатора). Затем среди всех пользователей проводится голосование за кандидатов, где вес каждого голоса определяется суммой активов голосующего. По результатам этого голосования выбирается некоторое число (*обычно 20-50*) кандидатов, которые получают право формировать новые блоки транзакций. Правила протокола гарантируют корректное принятие решений, если большая часть активов, принимающих участие в голосовании, контролируется «честными» пользователями.

Выбранные в результате голосования валидаторы перемешиваются случайным образом, образуя очередь. Перемешивание выполняется в соответствии со специальным алгоритмом, так что предсказать очередь практически невозможно. Далее выделяется период времени, в течении которого каждый из валидаторов должен сформировать один блок очереди. При

этом либо валидатор успевает проверить новые транзакции и сформировать новый блок на основании предыдущего, либо эту работу сделает валидатор, но уже следующий в очереди. После завершения установленного периода времени (обычно порядка 1 сек) валидаторы снова перемешиваются и формируют новую очередь.

Важно отметить, что держатели монет могут выполнять переголосование за кандидатов в произвольное время. Т.о. текущая группа валидаторов может измениться и их новая очередь будет сформирована уже другим составом. Кроме того, один держатель монет может голосовать более чем за одного кандидата, распределяя вес своих монет пропорционально между несколькими кандидатами.

Посредством DPoS пользователи могут выбирать репрезентативный узел для голосования от их имени, выступая в качестве доверенного лица для голосования. Делегированный узел выполняет такие задачи, как проверка подписей для блоков, которые обрабатываются, а в случае конфликтующих транзакций - голосование за действительную транзакцию.

Используется в:

- Steemit;
- EOS;
- BitShares [16].

Плюсы:

- дешевые транзакции;
- масштабируемость;
- энергоэффективный;
- возраст монеты не имеет значения: - отсутствие возраста означает, что перемещение монет менее затратно;
- результатом является стабильный, постоянный интерес только для активных кошельков и только с небольшими затратами;
- время простоя значительно влияют на пользовательский интерес к DPOS.

Минусы:

- подвержен атаке «Ничего на кону»;
- частично централизованный.

4.4 Proof-of-Stake-Time (PoST).

Это новый подход к формированию консенсуса путем введения компонента времени ставки, где вероятность ставок увеличивается со временем, то есть ставка времени является произведением общего количества монет (C) и доли (f) приемлемого возраста (a). Это улучшает стимул для размещения и повышает безопасность сети. Это также усиливает децентрализацию консенсуса, что значительно превосходит существующие стандарты.

Используется в: PostCoin и Vericoin.

Плюсы: - поддерживает эффективность Proof-of-Stake.

4.5 Proof of stake Boo (PoS Boo)

PoS Boo - это PoS-схема, основанная на PoS Casper. Эта схема Casper лучше всего подходит для «POSv3» с введением фактора риска для злоумышленников. Система является прогрессивной, что значительно затрудняет выполнение таких атак, как атака 51%. Вам понадобятся большинство всех монет, и вы также столкнетесь с возможностью потерять их все при запуске такой атаки. Завершенность в основном определяется ставкой и факторами риска.

Плюсы:

- трудно выполнить атаку успешно даже с 51%;
- обеспечивает цензуру транзакций. С PoW майнер блоков может не майнить блок, содержащий определенные адреса, тем самым подвергая цензуре этот сетевой адрес. Так как создатели блоков выбираются случайным образом и валидаторы являются глобальными

с этой схемой PoS, очень сложно подвергать цензуре адреса из сети (в итоге можно потерять свою ставку).

Используется в SHIELD.

4.6 High Interest Proof of Stake (HiPoS)

Возраст монет используется в расчете для веса ставки, но не для вознаграждения за ставку. Ставка вознаграждения фиксируется по расписанию. Результатом является стабильный, постоянный интерес для размещения кошельков, поскольку время простоя минимально и затраты невелики. Большой входной размер строго наказывается HiPOS.

Используется в: - Positron (2015); - BitBean (2015); - EdgeCoin (EDGE) [17].

Плюсы:

- стимулирует и содействует удержанию пользователей, поскольку разработчики выпускают все больше информации о своих новых проектах.
- позволяет участникам с меньшими ресурсами/запасами получить большую выгоду, просто найдя несколько блоков в нужное время.

4.7 Traditional Proof of stake / Tiered Proof Of Stake (TPOS)

Это форма алгоритма, с помощью которой криптовалютная сеть Blockchain стремится реализовать распределенное соглашение. В производных от TPOS валютах источник следующего блока выбирается с помощью различных комбинаций случайного сбора и ставки.

Используется в XSN [18].

Плюсы: платежи в большей части рассчитаны на держателей монет, а не на майнеров.

4.8 Casper the Friendly Finality Gadget (FFG)

Алгоритм:

1. Валидаторы ставят часть своих ресурсов в виде ставки.
2. Валидаторы начинают проверку блоков. При обнаружении блока, который, по их мнению, может быть добавлен в цепочку, осуществляется его проверка и делается ставка.
3. В случае если блок будет добавлен, валидаторы получают вознаграждение пропорционально их ставкам.

Плюсы:

- все преимущества Proof of stake.
- возможность «наказания» валидаторов, пытающихсяпровести атаку «ничего не поставлено на карту».
- возможность «наказания» майнеров, которые уходят из сети.

Используется в Casper the Friendly GHOST: Correct-by-Construction (CBC): полный PoS.

4.9 Proof of Stake Velocity (PoSV)

PoSV предлагается в качестве альтернативы PoW и PoS для защиты одноранговой сети и подтверждения транзакций Reddcoin, криптовалюты, созданной специально для облегчения социальных взаимодействий. PoSV предназначен для поощрения как владения (Stake), так и активности (Velocity), которые напрямую соответствуют двум основным функциям Reddcoin, как реальной валюты: - хранилищу стоимости и средству обмена. Reddcoin также может функционировать в качестве расчетной единицы в разном социальном контексте.

Используется в: Reddcoin [19].

4.10 Magi's proof-of-stake (mPoS)

Цели достижения распределенного консенсуса посредством операций в дополнение к mPoW. mPoS спроектирован таким образом, что он отклоняет потенциальные атаки путем накопления большого количества монет или времени автономной игры, что приводит к проблемам с безопасностью. По аналогии с mPoW, mPoS строится в соответствии с концепцией

модели притяжения-отталкивания. Magi является гибридным решением mPoW с mPoS и объединяет оба согласованных подхода, чтобы получить преимущества от двух механизмов и создать более надежную платежную систему.

Используется в: MAGI и Bitcointalk [20,21].

4.11 Leased Proof-of-Stake (LPoS)

Вероятность нахождения нового блока зависит только от того, сколько есть токенов (т.е. от ставки).

Используется в: NXT и Waves [14,15].

Плюсы: не требует значительной вычислительной мощности, чтобы создать новый блок.

4.12 Leasing Proof of Stake (PoS/LPoS)

LPoS - это расширенная версия Proof-of-Stake. В обычной системе Proof-of-Stake каждый узел, который содержит определенную сумму криптовалюты, имеет право добавить следующий блок в цепочку блоков, но в системе LPoS на платформе Waves пользователи могут сдавать в аренду свой баланс для полных узлов. С LPoS пользователь будет иметь возможность сдавать в аренду WAVES из кошелька разным подрядчикам, которые могут выплатить процент в качестве вознаграждения. Чем больше сумма, сдаваемая в аренду для полного узла, тем выше вероятность того, что этот узел будет выбран для создания следующего блока. Если этот полный узел выбран для создания следующего блока, арендатор получит процент от суммы транзакции, которую собирает полный узел. В среде LeasedProof-of-Stake пользователи могут выбирать между выполнением полного узла или передачей своей доли в полный узел с получением вознаграждений. Эта система позволяет любому участвовать в обслуживании сети Waves. Пользователь может передавать свои волны через лизинг на любом компьютере или мобильном устройстве, имеющем интернет-браузер, поскольку Waves предоставляет облегченное клиентское решение, не требующее майнеров, которые сдают в аренду свой баланс для хранения всей цепочки блоков или для запуска кошелька.

Используется в: NXT и Waves [14,15].

5 Выводы

Протоколы консенсуса являются неотъемлемой частью распределенных систем. В первую очередь они помогают достигать справедливости, избегать сбоев системы, когда один из участников (узлов) выходит из строя. Во-вторых, децентрализованная среда требует решения, которое поможет двигаться вперёд и изменять общее состояние, даже в среде, где никто никому не доверяет. Определенные правила помогают достичь «консенсуса».

В работе рассмотрена лишь некоторая часть, наиболее распространенных, алгоритмов проколов консенсуса, которые имеют разные характеристики, особенности и уязвимости. Сделан вывод о том, что, не существует идеального алгоритма, а у каждого из них есть свои плюсы и минусы. Используя их сильные стороны, в каждом конкретном случае, следует подбирать наиболее подходящий вариант реализации алгоритма, максимально учитывающий специфику условий и основной контекст решаемой задачи.

Ссылки

[1] Roper Klacks. Blockchain. "2018. URL: https://en.wikipedia.org/wiki/Blockchain
[2] Cedric Walter. Blockchain Consensus. URL: https://tokens-economy.gitbook.io/consensus/
[3] IOTA. URL: https://www.iota.org/
[4] Hegera. URL: https://www.hedera.com/
[5] Sybil attack. URL: https://en.wikipedia.org/wiki/Sybil_attack
[6] Nano. URL: https://nano.org/en
[7] LYNX. URL: https://getlynx.io/
[8] McConlogue P. Building a Blockchain Singularity with Proof of Distance.URL: https://blog.blockcollider.org/building-a-blockchain-singularity-with-proof-of-edit-distance-1d60c328de7a
[9] Форк. URL: https://ru.bitcoinwiki.org/wiki/%D0%A4%D0%BE%D1%80%D0%BA

[10] APR Coin. URL: https://apr-coin.com/
[11] Tezos. URL: https://tezos.com/
[12] Decred. URL:https://decred.org/ru/
[13] EOIS. URL: https://eos.io/
[14] NXT. URL: https://nxtplatform.org/
[15] Waves. URL: https://wavesplatform.com/
[16] Bitshares. URL: https://bitshares.org/
[17] EdgeCoin. URL: https://www.edgecoin.io/
[18] XSN. URL: https://stakenet.io/
[19] Reddcoin. URL: https://reddcoin.com/
[20] MAGI. URL: https://www.m-core.org/
[21] Bitcointalk. URL: https://bitcointalk.com/

Reviewer: Alexandr Potii, Dr. of Sciences (Engineering), Full Prof., V. N. Karazin Kharkiv National University, Kharkiv, 61022, Ukraine. E-mail: potav@ua.fm

Received: March 2019.

Authors:
Diana Kovalchuk, student of CSD, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
E-mail: dianakovalhyk@ukr.net
Tetiana Ivko, Researcher, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
E-mail: t.ivko@outlook.com
Tetiana Kuznetsova, Researcher, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
E-mail: kuznetsova.tatiana17@gmail.com
Oleksii Nariezhnii, Ph.D., Associate Professor, V.N. Karazin Kharkiv National University, Ukraine.
E-mail: o.nariezhnii@karazin.ua

Review of consensus protocols used in Blockchain technologies.

Abstract. The categories of popular consensus punctures are reviewed: Proof of Work and its hybrids, Proof of Stake (includes LPoS, Proof of Importance) and hybrids, DAGs and its varieties. The article describes their algorithms, characteristics, features, as well as disadvantages and advantages. Vulnerabilities and attacks are subject to. Also provides a list of the use of each protocol in cryptocurrencies and other systems.

Keywords: Consensus; Blockchain; Blockchain Consensus Protocol; Decentralized Systems; Distributed Registry.

Рецензент: Олександр Потій, д.т.н., проф., Харківський національний університет імені В.Н. Каразіна, Харків, 61022, Україна.
E-mail: potav@ua.fm

Надійшло: Березень 2019.

Автори:
Діана Ковальчук, студентка факультету комп'ютерних наук, Харківський національний університет імені В.Н. Каразіна, Харків, 61022, Україна.
E-mail: dianakovalhyk@ukr.net
Тетяна Івко, науковий співробітник, Харківський національний університет імені В.Н. Каразіна, Харків, 61022, Україна.
E-mail: t.ivko@outlook.com
Тетяна Кузнецова, науковий співробітник, Харківський національний університет імені В.Н. Каразіна, Харків, 61022, Україна.
E-mail: kuznetsova.tatiana17@gmail.com
Олексій Нарєжній, к.т.н., доцент кафедри, Харківський національний університет імені В.Н. Каразіна, Харків, 61022, Україна..
E-mail: o.nariezhnii@karazin.ua

Огляд протоколів консенсусу, що застосовуються в технологіях блокчейн.

Анотація. Розглянуто категорії популярних проколів консенсусу: Proof of Work і його гібриди, Proof of Stake (LPoS, Proof of Importance) та гібриди, DAGs. У роботі стисло описані їх алгоритми, характеристики, особливості, а також недоліки та переваги. Зазначені відомі вразливості та атаки до яких вони схильні. Наведено приклади використання кожного з протоколів, що розглядаються, в криптовалютах та інших системах.

Ключові слова: консенсус; блокчейн; протокол консенсусу блокчейн; децентралізовані системи; розподілений реєстр.

УДК 004.738.5 DOI: 10.26565/2519-2310-2019-1-04

ЗАСТОСУВАННЯ КРИПТОАЛГОРИТМІВ В ДЕЦЕНТРАЛІЗОВАНИХ МЕРЕЖАХ ТА ПЕРСПЕКТИВИ ЇХ ЗАМІНИ ДЛЯ ПОСТКВАНТОВОГО ПЕРІОДУ

Олександр Якішин, Олександр Оникійчук, Володимир Скриннік, Катерина Кузнецова

Харківський національний університет імені В.Н. Каразіна, майдан Свободи 4, Харків, 61022, Україна
dkrakaslian@gmail.com, onik4524a@gmail.com, v.skrynnik@karazin.ua, kate7smith12@gmail.com

Рецензент: Олександр Оксіюк, д.т.н., проф., Київський національний університет імені Т. Шевченка, вул. М. Ломоносова 81, Київ, 03189, Україна.
o.oksiuk@gmail.com

Надійшло: Березень 2019.

Анотація: *В роботі проведено огляд використовуваних у блокчейн системах електронних підписів та функцій хешування. Наведено криптографічні алгоритми, які використовуються або можуть використовуватися в децентралізованих мережах. Представлені алгоритми для захисту, як від класичних, так і від квантових атак. Результати роботи свідчать про необхідність вдосконалення криптографічних алгоритмів для захисту від можливих квантових атак в майбутньому.*

Ключові слова: *криптографічні алгоритми; алгоритми захисту децентралізованих систем; блокчейн.*

1 Вступ

Нині досить широке розповсюдження знаходять технології блокчейн, в цих технологіях особливо жорсткі вимоги висувають до надання користувачам таких послуг безпеки, як цілісність, справжність, доступність. Ці вимоги задовольняються використовуючи у цих системах алгоритм електронного підпису (ЕП) та функції гешування (ФГ). ЕП у блокчейні, в основному, використовується для підпису транзакцій та, у окремих випадках (таких як Etherium), для виконання смарт-контракту, при цьому основними вимогами до електронного підпису є забезпечення кріптостійкості проти класичних та квантових атак. Одночасно, сутність використання ФГ полягає у застосуванні її до обчислення геш значень блоків та зв'язування ланцюгів блоків, при цьому основними вимогами до геш-функцій є забезпечення криптостійкості проти класичних та квантових атак, а також мінімізації складності (підвищення швидкості) гешування. Метою цієї роботи є загальний аналіз вже використовуваних у блокчейн системах ЕП та ФГ і огляд перспективних ЕП та ФГ, які у майбутньому можуть їх замінити.

2 Визначення блокчейн

Блокчейн (*англ. Blockchain, Block chain від block – блок, chain – ланцюг*) [1] – це незмінні системи цифрових регістрів, які реалізовані розподіленим чином (тобто, без центрального місця сховища) та зазвичай без центрального органу управління. На самому базовому рівні вони дозволяють користувачам записувати транзакції в загальнодоступному реєстрі, так що ніяка транзакція не може бути змінена після її опублікування.

Транзакція - це запис передачі активів (цифрова валюта, одиниці запасів, тощо) між сторонами. Аналогом цього було б запис на перевірочному рахунку кожного разу коли гроші було депоновано або вилучено. У Таблиці 1 наведено умовний приклад транзакції. Кожен блок в блокчейні містить кілька транзакцій. Одна транзакція зазвичай вимагає принаймні наявність наступних інформаційних полів, але може містити більше:

• Сума - загальна сума переданого цифрового активу.

• Вхід - список цифрових активів, що підлягають передачі (їх загальна вартість дорівнює сумі). Слід зверніть увагу на те, що кожен цифровий актив ідентифікується унікально і може мати різні значення інших активів. Однак при цьому активи не можуть бути додані або вилучені з існуючих цифрових активів. Замість цього, цифрові активи можуть бути розділені на

кілька нових цифрових активів (кожен з меншою вартістю) або об'єднані, щоб сформувати меншу кількість нових цифрових активів (кожен з відповідним чином більшим значенням).

• Виходи - рахунки, які будуть одержувачами цифрових активів. Кожен вихід вказує значення, яке буде передано новому власнику (власникам), особу нового власника (ів), і набір умов, яким повинні відповідати нові власники, щоб отримати цю вартість. Якщо наданих цифрових активів більше, ніж потрібно, додаткові кошти повертаються відправнику (це механізм “внесення змін”).

ID транзакції / Hash - унікальний ідентифікатор кожної транзакції. Деякі блокчейн використовують ідентифікатор, а інші приймають геш конкретної транзакції, як унікальний ідентифікатор.

Таблиця 1 - Приклад транзакції

ID транзакції / Hash	Вхід	Виходи	Вага	Сума
Ідентифікатор транзакції: 0xa1b2c3	Рахунок A	Рахунок B	0.0321	
		Рахунок C	2.5000	2.5321

Важливим є визначення дійсності угоди (*той факт, що хтось претендує на те, щоб угода відбулася, ще не означає, що вона дійсно сталася*). Транзакції підписуються і їх можна перевірити за допомогою пар публічних/приватних ключів у будь-який час.

3 Геш

Важливою складовою технології блокчейн є використання для багатьох операцій криптографічних геш-функцій, наприклад таких як гешування змісту блоку. Гешування - це метод обчислення відносно унікального виводу фіксованого розміру (називається дайджест повідомлення або просто дайджест) для входу майже будь-якого розміру (наприклад, файлу, тексту або зображення). При цьому, навіть найменша зміна вхідного значення (наприклад, одного біта) призведе до абсолютно іншого дайджесту виводу (див. Табл. 2).

В цілому геш-алгоритми розроблені так, щоб вони були односторонніми (відомі як стійкі до показу): це означає, що неможливо знайти обчислювальну інформацію для будь-якого входу, який відображається на будь-якому заздалегідь заданому виході. Якщо треба знайти бажаний конкретний висновок, багато входів повинні бути випробувані шляхом передачі їх через геш-функцію, поки не буде знайдено вхідний сигнал, який дасть бажаний результат. Крім того, геш алгоритми розроблені таким чином, щоб бути стійкими до зіткнень (відомі як другий стійкий до показу).

Таблиця 2 - Приклади вхідного значення та SHA-256 вихідного значення

Вхідне значення	SHA-256 вихідне значення
1	0x6b86b273ff34fce19d6b804eff5a3f5747ada4eaa22f1d49c01e52ddb7875b4b
2	0xd4735e3a265e16eee03f59718b9b5d03019c07d8b6c51f90da3a666eec13ab35
Hello, World!	0xdffd6021bb2bd5b0af676290809ec3a53191dd81c7f70a4b28688a362182986f

Наприклад алгоритм гешування, що використовується у багатьох технологіях блокчейн, є Secure Hash Algorythm (SHA) з вихідним розміром 256 біт (SHA-256). Багато комп'ютерів апаратно підтримують цей алгоритм, що робить дуже зручним для обчислення.

Оскільки ФГ має дуже велику кількість можливих вхідних значень та безкінечну кількість вихідних значень, то можливо знайти колізію де $hash(x) = hash(y)$. Однак на даний момент це малоймовірно тому, що будь-який такий вхід x та y, який виробляє один і той самий дайджест, був би дійсним у контексті системи блокчейн (в даному випадку це були би обидві дійсні транзакції блокчейн), а також обчислювався б досить близько один до одного по часу. У переважної більшості сучасних комп'ютерів занадто мала обчислювальна швидкість для роботи з існуючими алгоритмами, що націлені на виконання подібних задач. В цьому контексті стоїть відмітити, що ще з 1980-го року почалась розробка квантового комп'ютера, який повинен значно перевершити по своїй обчислювальної потужності традиційні комп'ютери за допомогою обчислень що базуються на законах квантової механіки. Згідно з даними інтерв'ю з менеджера компанії IBM® Марком Кетченом, створення квантового комп'ютера прогнозується через 3-8 років [2]. На сьогодні існують два алгоритма для квантового комп'ютера, це алгоритм Гровера [3], який вирішує проблему перебору, та алгоритм Шора [4] що вирішує проблему факторизації (*розкладу одного числа на два*). З цих двох квантових алгоритмів для пошуку колізій може використовуватись алгоритм Гровера. Алгоритм Шора, на даний момент, ніяк не загрожує ФГ.

Розглянемо найпоширеніші ФГ у блокчейн системах:

1. SHA-256 (англ. *Secure Hash Algorithm Version 2* – безпечний алгоритм гешування, вер. 2), це збірна назва односторонніх геш-функцій SHA-224, SHA-256, SHA-384 і SHA-512 [5]. Найпоширеніша ФГ завдяки своїй простоті обчислення (високій швидкодії) та доволі непоганою стійкістю до колізій та пошуку першого прообразу, та має один основний недолік це невелика у порівнянні з іншими найпоширенішими ФГ стійкість до знаходження другого прообразу, що зазвичай і стає причиною використання сторонніх ФГ на вихідне значення SHA-256. Наприклад, у Bitcoin вихідне значення SHA-256 гешується спочатку RIPMD-160, потім BASE-58 і вже після цього використовується у системі блокчейн. На даний момент SHA-256 використовується у таких блокчейн системах як: Bitcoin, Litecoin, Tether, Cornado (*HMAC-SHA512*) і т.д.

2. Keccak (SHA-3) – геш функція побудована за принципом криптографічної губки [6] (*дана структура криптографічних алгоритмів була запропонована авторами алгоритму Keccak раніше*). Він є переможцем останнього на сьогодні конкурсу NIST SHA-3, де відрізнився від конкурентних ФГ підвищеною стійкістю до усіх видів атак, але не найбільшою швидкодією. Оригінальний алгоритм Keccak має безліч параметрів, що налаштовуються з метою забезпечення оптимального співвідношення криптостійкості і швидкодії для певного застосування алгоритму на певній платформі. Регульованими величинами є: розмір блоку даних, розмір стану алгоритму, кількість раундів у функції *f()* та інші. На даний момент Keccak використовується у таких блокчейн системах як Ethereum, Nexus(NXS), Quark(QRK), SmartCash(SMART), Maxcoin(MAX), CreativeChain(CREA).

Таблиця 3 – Порівняння параметрів алгоритмів SHA-256 та Keccak (SHA-3)

Алгоритм	Розмір виводу (біт)	Розмір внутрішнього стану	Розмір блоку	Довжина розміру	Розмір слова	Раунди
SHA -224 -256	256	256	512	64	32	64
SHA-384, -512	384/512	512	1024	128	64	80
Keccak (SHA-3)	224/256/384 /512	1600	1600-2*bits (1152/1088/ 832/576)	–	64	24

Аналіз даних що наведені в Табл. 3 [7], дозволяє стверджувати, що кожний алгоритм має свій розмір виводу, що варіюється від 256 до 512 біт, а найбільший розмір внутрішнього ста-

ну має алгоритм Keccak (SHA-3) – 1600, довжина розміру алгоритму варіюється від 64 до 128, а розмір слова (в залежності від алгоритму) від 32 до 64.

З аналізу даних Табл. 4 [8] можна зробити висновок, що 512 версії алгоритмів є кращими за своїх попередників, оскільки мають найбільшу стійкість до: - колізій; - знаходження прообразу; - знаходження другого прообразу та до алгоритму Гровера. Найкращим можна назвати ФГ SHA-3 512, оскільки має найкращу стабільність та стійкість, а найгіршою - ФГ SHA-256.

Таблиця 4 – Порівняння стійкості ФГ (у бітах)

Функції гешування	Стійкість до колізій	Стійкість до знаходження прообразу	Стійкість до знаходження 2-го прообразу	Стійкість до кв. алгоритму Гровера
SHA-256	128	256	201-256	64
SHA-512	256	512	394-512	128
SHA-3 256	128	256	256	64
SHA-3 512	256	512	512	128

Важливим критерієм для порівняння ФГ є складність/швидкодія хешування [8], що наведені у Табл. 5.

Таблиця 5 - Результати тесту швидкодії ФГ

Алгоритм	MiB/sec	Циклів/байт
SHA-3 256	46	6,2
SHA-256	111	15,8
SHA-512	99	17,7
SHA-3 512	67	5,8

З даних Табл. 5 можна побачити, що алгоритм SHA-256 має найбільший показник MiB/sec – 111, SHA-3 256 – найменший, 46. ФГ SHA-512 виконує 17,7 циклів за байт, а SHA-3 512 найменше – 5,8. Таким чином, серед наведених алгоритмів, SHA-256 та SHA-512 – найшвидші ФГ.

В цілому за результатами аналізу даних всіх вище наведених таблиць можна зробити висновок, що сучасні функції гешування дозволяють забезпечити необхідний рівень стійкості проти усіх відомих класичних атак. Однак, варто зазначити, що з вхідною довжиною у 256 біт стійкість до алгоритму Гровера не перевищує 64 біт, що не має необхідної стійкості для постквантового періоду. Хоча за результатами тесту швидкості найкращим є SHA-2 варто не забувати, що наведені результати стосуються тільки програмної реалізації цих ФГ, швидкодія яких відрізняється від реалізації цих ФГ на мікроконтролерах, також SHA-2 у порівнянні з більшістю інших приведених функцій гешування відрізняється меншою стійкістю до атак знаходження другого прообразу.

4 Криптографія асиметричного ключа

Фундаментальною технологією, що використовується технологіями блокчейн, є *криптографія асиметричних ключів* (інша назва криптографія публічного/приватного ключа). Криптографія асиметричного ключа використовує пару ключів: відкритий ключ і приватний ключ, які математично пов'язані між собою. Відкритий ключ може бути оприлюднений без зниження безпеки процесу, але приватний ключ повинен залишатися таємним, якщо дані повинні зберігати криптографічний захист. Навіть при наявності взаємозв'язку між двома ключами, це не дає можливості ефективно визначити приватний ключ на основі знання відкритого ключа. Криптографія асиметричного ключа використовує різні ключі з пар ключів для конкретних функцій, залежить від того, яка послуга повинна бути надана. Наприклад, при цифро-

вому (електронному – ЕП) підпису даних, криптографічний алгоритм використовує приватний ключ для підпису. Потім підпис можна перевірити за допомогою відповідного відкритого ключа.

Використання криптографії асиметричного ключа в системах блокчейн:

• приватні ключі використовуються для цифрового підпису транзакцій;

• відкриті ключі використовуються для виведення адрес, що дозволяє використовувати підхід один-до-багатьох, що запроваджує псевдоанімність (одна пара відкритих ключів може дати кілька адрес);

• відкриті ключі використовуються для перевірки підписів, створених за допомогою приватних ключів;

• криптографія асиметричного ключа дає можливість перевірити, що користувач передає значення іншому користувачеві, що володіє приватним ключем, здатним підписати значення.

Деякі блокчейн системи, наприклад такі як Monero, використовують ЕП у протоколі кільцевого підпису. Кільцевий підпис - один з механізмів реалізації електронного підпису, при якому відомо, що повідомлення підписав один з членів списку потенційних підписантів, але не розкриває, хто саме. Підписант самостійно формує список з довільного числа осіб (включаючи і себе). Для накладання підпису підписувачу не потрібен дозвіл, сприяння або допомога з боку включених у список осіб, використовуються лише відкриті ключі всіх членів списку і власний закритий ключ.

Варто позначити, що у більшості алгоритмів ЕП криптостійкість базується на складності вирішення проблеми факторизації, що робить загрозою не тільки алгоритм Гровера, але й алгоритм Шора.

Розглянемо деякі найпоширеніші алгоритми ЕП застосовані у системах блокчейн [9].

ECDSA [10] - алгоритм з відкритим ключем для створення цифрового підпису, аналогічний за своєю будовою DSA, але визначений, на відміну від нього, не над кільцем цілих чисел, а в групі точок еліптичної кривої. Стійкість алгоритму шифрування ґрунтується на задачі дискретного логарифма в групі точок еліптичної кривої. На відміну від задачі простого дискретного логарифма і задачі факторизації цілого числа, не існує субекспоненціального алгоритму для задачі дискретного логарифма в групі точок еліптичної кривої. З цієї причини «сила на один біт ключа» істотно вище в алгоритмі, який використовує еліптичні криві. Крім того, ECDSA використовує не тільки алгебраїчні властивості еліптичних кривих, але і кінцеві поля. Кінцеве поле - це заданий діапазон додатних чисел, в рамках якого лежать результати алгебраїчних обчислень. Еліптичні криві в рамках кінцевого поля змінюються до невпізнання. Але, незважаючи на втрачену «красу», все математичні властивості кривої залишаються колишніми. Для підписування повідомлень необхідна пара ключів - відкритий і закритий. При цьому закритий ключ повинен бути відомий тільки тому, хто підписує повідомлення, а відкритий - будь-кому хто бажає перевірити справжність повідомлення. Також загальнодоступними є параметри самого алгоритму. ECDSA є дуже привабливим алгоритмом для реалізації цифрового підпису.

Найважливішою перевагою ECDSA є можливість його роботи на значно менших полях $F(p)$. Як, загалом, з криптографією еліптичної кривої, передбачається, що бітовий розмір відкритого ключа, який буде необхідний для ECDSA, дорівнює подвійному розміру секретного ключа в бітах. Для порівняння, при рівні безпеки в 80 біт (тобто атакуючому необхідно приблизно 2^{80} версій підписів для знаходження секретного ключа), розмір відкритого ключа DSA дорівнює, принаймні, 1024 біт, тоді як відкритого ключа ECDSA - 160 біт. З іншого боку розмір підпису однаковий і для DSA, і для ECDSA: $4 \cdot t$ біт, де t - рівень безпеки, який вимірюється в бітах, тобто - приблизно 320 біт для рівня безпеки в 80 біт. У мережі Bitcoin ECDSA використовується зі спеціальними параметрами означеними, як secp256k, secp256k1, він майже ніколи не використовувався до того, як Bitcoin став популярним, але зараз він набуває популярності завдяки своїм кількома приємним властивостям. Найбільш часто використовувані криві мають випадкову структуру, але secp256k1 був побудований спеціальним невипадковим способом, який дозволяє особливо ефективно обчислювати. Як наслідок, він ча-

сто більш ніж на 30% швидше, ніж інші криві, якщо реалізація достатньо оптимізована. Крім того, на відміну від популярних кривих NIST, константи secp256k1 були обрані передбачуваним способом, що значно зменшує можливість створення творцем кривої будь-якого типу бекдора. На сьогодні ECDSA використовується у більшості децентралізованих систем, таких як: Bitcoin, Ethereum, Litecoin та інші.

З даних таблиці 6 видно, наскільки алгоритм ЕП ECDSA кращий за DSA [11], тому що маючи однаковий рівень безпеки, розмір ключа алгоритму ECDSA набагато менший за свого конкурента, а це може стати перевагою у програмах, в яких є обмеження в реальному часі та пам'яті.

Таблиця 6 - Порівняння розміру ключа

Безпека (у бітах)	DSA – розмір ключа	ECDSA – розмір ключа
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

Так як у більшості алгоритмів ЕП криптостійкість базується на складності вирішення проблеми факторизації, що робить загрозою не тільки алгоритм Гровера, але й алгоритм Шора, має сенс розглянути і перспективні алгоритми ЕП, які у майбутньому можуть замінити нинішні. А саме, учасники що пройшли у другий раунд конкурсу NIST Post-Quantum crypto Project [12] такі, як: - Crystals Dilithium; - SPHINCS+ та SPHINCS; - XMSS.

1. Crystals Dilithium – алгоритм ЕП який базується на методі "Fiat-Shamir з переривання-ми", який використовує вибірку відхилення, для того, щоб зробити схеми Fiat-Shamir на основі решіток компактними та безпечними. Схема з найменшими розмірами підпису з використанням такого підходу – це схема Ducas, Durmus, Lepoint та Lyubashevsky, що базується на припущенні NTRU і вирішальним чином використовує гаусовську вибірку для створення підписів. Оскільки вибірку з Гауса важко здійснити безпечно та ефективно, вирішено використовувати лише рівномірний розподіл [13]. Dilithium покращує найефективнішу схему, яка використовує тільки рівномірний розподіл, завдяки Bai та Galbraith, використовуючи новий метод, який скорочує відкритий ключ більш ніж у 2 рази. Наскільки відомо, Dilithium має найменший відкритий ключ та розмір підпису у порівнянні з будь-якою схемою підпису на основі решітки, яка використовує лише рівномірну вибірку.

2. SPHINCS+ – оптимізований варіант SPHINCS криптографічної схеми, заснованої на криптографічній стійкості геш-функції. Вона включає в себе кілька поліпшень, спеціально спрямованих на зменшення розміру підпису. Схема включає в себе алгоритм формування та перевірки електронного цифрового підпису.

3. SPHINCS – представляє з себе високонадійну систему цифрового підпису [14], що заснована на геш-функції без збереження даних, яка представляє з себе безпеку «128 bit» проти атак квантового комп’ютеру (Табл. 7-8). Як свідкує аналіз, механізми ЕП класу SPHINCS, розроблені на достатньому рівні, при зрозумілих моделях загроз, в том числі за умов криптографічної стійкості до атак на основі стійкості до знаходження прообразу та стійкості до колізій. Оптимізаційні рішення SPHINCS полягають у тому, щоб включити у підпис усі вузли на певному рівні. Це дозволить уникнути зберігання автентифікаційних шляхів над тим рівнем. SPHINCS представляє дві нові ідеї, які суттєво зменшують розмір підпису.

По-перше, щоб підвищити рівень безпеки випадкового вибору індексу, SPHINCS замінює лист OTS схемою геш-підпису (FTS). FTS є, як випливає з назви, схемою підпису, яка призначена для підписання декількох повідомлень; в контексті SPHINCS це дозволяє кілька колізій індексів, що, в свою чергу, дозволяє зменшити висоту дерев для того ж рівня безпеки. Наприклад, SPHINCS-256 зменшує загальну висоту дерева з 256 до всього 60, зберігаючи при цьому захист від квантових нападників.

На відміну від SPHINCS, SPHINCS+ був кілька покращений. По-перше, був змінений захист від багатоцільових атак. При цьому, основна концепція полягає у використанні різної геш-функції для кожного виклику. Кожен виклик геш-функції виконується за допомогою іншого ключа і застосовує різні бітові маски. Ключі і бітові маски генеруються псевдовипадково з адреси, що вказує контекст виклику, і публічного нащадку. Щоб отримати абстракцію, були введені поняття налаштованих геш-функцій, які крім вхідного значення приймають адресу публічного нащадку.

Таблиця 7 – Розмір алгоритму SPHINKS

Підпис	41000 bytes
Розмір публічного ключа	1056 bytes
Розмір приватного ключа	1088 bytes

Таблиця 8 – Параметри алгоритму SPHINKS

Значення	Параметр
256	Довжина біта хешів у HORST і WOTS
512	Довжина біта хеша повідомлення
60	Висота гіпер-дерева
12	Шари гіпер-дерева
16	Параметр Winternitz, що використовується для підписів WOTS
2^{16}	Кількість секретних елементів HORST
32	Кількість виявлених елементів секретного ключа в кожному сигналі HORST

По-друге, стиснення відкритого ключа WOTS + відбувається з відкритим ключем: останні вузли ланцюжків WOTS + стискаються не за допомогою L-дерева, а за допомогою одного настроюваного виклику геш-функції. Цей виклик знову отримує адресу і загальнодоступне початкове значення для введення цього виклику і для генерації бітової маски протягом усього часу введення. Це було неможливо раніше, не підірвавши відкритий ключ. Тепер, коли бітові маски генеруються псевдовипадково, це не впливає на розмір відкритого ключа.

По-третє, HORST схеми з короткою підписом були замінені на FORS схеми. Перевага полягає в тому, що тепер є можливість використовувати набагато менші параметри і, тим самим, в кінцевому підсумку виграти в розмірі та швидкості підпису. Остання зміна стосується вибору індексу що перевіряється.

В SPHINCS пара ключів HORST, використовувана для підпису повідомлення, була обрана, генеруючи індекс псевдовипадковим чином. Оскільки задіяно секретне початкове значення, верифікатор не зміг перевірити, чи був цей індекс дійсно згенерований таким чином. Це мало недолік, що полягає в тому, що зловмисник, націлений на HORST, міг виконати багатоцільову атаку, використовуючи одне геш-обчислення для одночасного націлювання на всі екземпляри HORST. Це більше неможливо, оскільки кожне повідомлення, яке намагається згенерувати противник, тепер безпосередньо пов'язується з екземпляром FORS та стає непридатним для будь-якого іншого примірника. Крім того, це дозволяє опустити індекс в сигнатурі SPHINCS +. Ці зміни дозволили визначити набори параметрів з розмірами підпису.

4. XMSS – ЕП заснований на схемі підпису Меркла [15] і узагальненій схемі підпису Меркла (GMSS) [16]. XMSS є ефективною схемою пост-квантового підпису з мінімальними припущеннями безпеки. XMSS є ЕП подібним до SPHINCS, однак із більш коротшими підписами, за рахунок того, що вона фіксує дані. Наприклад, варіант XMSS-T генерує 8,8 кБ ЕП для продукування 2^{60} повідомлень та забезпечує 128-бітний квантовий захист. Головною відмінністю від SPHINCS є те, що гіпердерево SPHINCS ділиться на багато рівнів, так як ці дерева для кожного підпису мають в реальному часі повторно обчислюватися. Однак XMSS має пе-

ревагу за рахунок додаткового ефективного алгоритму, що дає змогу зекономити на вартості обчислення за рахунок створення багатьох підписів.

XMSS екзистенційно непідробний під прицільно обраних атак повідомлень в стандартній моделі. Схема XMSS дозволяє розширювати одноразові підписи [17]. Вимоги безпеки для XMSS мінімальні. Існування безпечної схеми підпису передбачає існування другого сімейства провізних стійких геш-функцій та сімейства псевдовипадкових функцій. XMSS є практичним, оскільки існує багато способів створити дуже ефективні (геш) сім'ї функцій, які, як вважають, є стійкими прообразами або псевдовипадковими, навіть за наявності квантових комп'ютерів. Наприклад, криптографічні геш-функції і блокові шифри можуть бути використані для цілей побудови таких сімей. Зокрема, існують такі конструкції, що базуються на складних задачах теорії алгебри та кодування. При цьому величезна кількість інстанцій XMSS гарантує довгострокову доступність надійних і ефективних схем підпису [18].

5 Висновки

В зв’язку з інтенсивним розвитком та впровадженням систем що реалізують децентралізований принцип управління, зросла потреба щодо забезпечення потрібних рівнів безпеки при використанні цих систем. В роботі розглянуті існуючі алгоритми захисту деяких криптографічних систем, зокрема алгоритми електронного підпису та функцій гешування, визначені мета і причина їх використання, вплив на відповідні системи, та на безпеку блокчейн в цілому. Наведені приклади найпоширеніших функцій гешування, а саме алгоритм SHA-256 та Keccak (він же SHA-3).

У роботі розглянута криптографія асиметричного шифрування, яка використовує відкриті та приватні ключі. Досліджено алгоритм ECDSA, який є алгоритмом з відкритим ключем для створення цифрового підпису, аналогічний за своєю побудовою до DSA. Крім того, проведено загальний аналіз та огляд перспективних електронних підписів і функцій гешування, які можуть замінити вже існуючі зразки. До таких слід віднести наступні:

- Crystals Dilithium – алгоритм, що базується на методі "Fiat-Shamir з перериваннями";
- SPHINCS+ – є оптимізованим варіантом криптографічної схеми SPHINCS;
- XMSS – ЕП, що заснований на схемі підпису Меркла.

У роботі розглядаються причини можливого використання даних алгоритмів у майбутньому, та аналізується їх стійкість перед «квантовими атаками», що є важливим аспектом існування та безпечного функціонування децентралізованих систем у майбутньому.

Посилання

[1] NISTIR 8202 Blockchain Technology Overview / Yaga D., Mell P., Roby N., Scarfone K. P. 50-59. URL: https://csrc.nist.gov/publications/detail/nistir/8202/final

[2] Mearian L. IBM Touts Quantum Computing Advance. URL: https://www.computerworld.com/article/2502275/ibm-touts-quantum-computing-advance.html

[3] Zalka Chr. Grover’s quantum searching algorithm is optimal. *Phys.Rev.*1999. A60. P. 2746 – 2751

[4] Shor P. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. *Foundations of Computer Science*. Proceedings 35th Annual Symposium on - IEEE. 1994. P.124 – 134.

[5] SHA-2. URL: https://ru.wikipedia.org/wiki/SHA-2

[6] Почему Keccak настолько крут и почему его выбрали в качестве нового SHA-3. URL: https://habr.com/ru/post/168707/

[7] Comparison of cryptographic hash functions. URL: https://en.m.wikipedia.org/wiki/Comparison_of_cryptographic_hash_ functions?wprov=sfti1

[8] NISTIR 7896. Third-Round Report of the SHA-3 Cryptographic Hash Algorithm Competition. *NIST*. 2012. P. 50 – 65.

[9] Digital Signature in Blockchain. URL: https://dzone.com/articles/digital-signature-2

[10] Johnson D., Menezes A., Vanstone S. The Elliptic Curve Digital Signature Algorithm (ECDSA). *International Journal of Information Security*. 2001. Vol. 1, Issue 1. P. 36 – 63.

[11] Haddaji R., Ouni R., Bouaziz S., Mtibaa A. Comparison of Digital Signature Algorithm and Authentication Schemes for H.264 Compressed Video (IJACSA). *International Journal of Advanced Computer Science and Applications*. 2016. Vol. 7, № 9. P.7.

[12] Post-Quantum Cryptography. URL: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions

[13] CRYSTALS-Dilithium Algorithm Specifications and Supporting Documentation / Ducas L. and all. URL: https://pq-crystals.org/dilithium/data/dilithium-specification.pdf

[14] SPHINCS: practical stateless hash-based signatures/ Bernstein D.J. and all. URL: https://www.iacr.org/archive /eurocrypt2015/90560214/90560214.pdf

[15] Merkle R. A certified digital signature. *Advances in Cryptology - CRYPTO’ 89* : *Proceedings* /Gilles Brassard, ed. Vol. 435 of Lecture Notes in Computer Science. Berlin: Springer /Heidelberg, 1990. P. 218–238.

[16] Merkle signatures with virtually unlimited signature capacity/ Buchmann J. and all. *Applied Cryptography and Network Security*/ Katz J. and Yung M. ed.Vol. 4521 of Lecture Notes in Computer Science. Berlin: Springer / Heidelberg, 2007. P.31–45.
[17] Rogaway P., Shrimpton T. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. *FSE*/ Roy B.K., Meier W., ed. Vol. 3017 of Lecture Notes in Computer Science. Berlin: Springer / Heidelberg, 2004. P. 371–388.
[18] Buchmann J., Dahmen E., Hülsing A. XMSS - A Practical Forward Secure Signature Scheme Based on Minimal Security Assumptions. *Post-Quantum Cryptography. PQ Crypto 2011*. Vol. 7071 of Lecture Notes in Computer Science. Berlin: Springer / Heidelberg, 2011. P. 117–129.

Reviewer: Oleksandr Oksiiuk, Doctor of Sciences (Engineering), Full Professor, Taras Shevchenko National University of Kiev 81 Lomonosova St., Kyiv, 03189, Ukraine. E-mail: o.oksiuk@gmail.com

Received on March 2019.

Authors:
Oleksandr Yakishin, computer science student, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
E-mail: dkrakaslian@gamil.com
Oleksandr Oniichichuk, computer science student, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
E-mail: onik4524a@gmail.com
Volodymyr Skrynnik, Researcher, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
E-mail: v.skrynnik@karazin.ua
Kateryna Kuznetsova, computer science student, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
E-mail: kate7smith12@gmail.com

Application of cryptographic algorithms in decentralized networks and prospects for their replacement for the post-quantum period.

Abstract. The article reviewed cryptographic algorithms that are used or can be used in decentralized networks. Algorithms for protection, both from classical and from quantum attacks are given. In the work there is a general analysis of the methods already used. Reviewed algorithms for protection, both from classical and from quantum attacks are given. The results of this work show the need for cryptographic algorithms to protect against possible quantum attacks in the future.

Keywords: Cryptographic algorithms; Algorithms for the protection of decentralized systems; Blockchain.

Рецензент: Александр Оксиюк, д.т.н., проф., Киевский национальный университет имени Т. Шевченко, ул. Ломоносова 81, Киев, 03189 Украина. E-mail: o.oksiuk@gmail.com

Поступила: Март 2019.

Авторы:
Александр Якишин, студент факультета компьютерных наук, Харьковский национальный университет имени В.Н. Каразина, Харьков, Украина. E-mail: dkrakaslian@gmail.com
Александр Оникийчук, студент факультета компьютерных наук, Харьковский национальный университет имени В.Н. Каразина, Харьков, Украина. E-mail: onik4524a@gmail.com
Владимир Скрынник, научный сотрудник, Харьковский национальный университет имени В.Н. Каразина, Харьков, Украина. E-mail: v.skrynnik@karazin.ua
Екатерина Кузнецова, студентка факультета компьютерных наук, Харьковский национальный университет имени В.Н. Каразина, Харьков, Украина. E-mail: kate7smith12@gmail.com

Применение криптоалгоритмов в децентрализованных сетях и перспективы их замены в постквантовый период.

Аннотация. В работе проведен обзор используемых в блокчейн системах электронных подписей и функций хеширования. Представлены криптографические алгоритмы, которые уже используются или могут использоваться в децентрализованных сетях. Приведены алгоритмы для защиты, как от классических, так и от квантовых атак. Результаты работы свидетельствуют о необходимости совершенствования криптографических алгоритмов для защиты от возможных квантовых атак в будущем.

Ключевые слова: криптографические алгоритмы; алгоритмы защиты децентрализованных систем; блокчейн.

УДК 004.738.5 DOI: 10.26565/2519-2310-2019-1-05

ПОБУДОВА СИСТЕМИ ГОЛОСУВАННЯ З ВИКОРИСТАННЯМ БЛОКЧЕЙН ТЕХНОЛОГІЙ НА ПРИКЛАДІ HYPERLEDGER

Микита Гончаров, Євген Деменко, Микола Полуяненко, Володимир Шлокін

Харківський національний університет імені В.Н. Каразіна, майдан Свободи 4, Харків, 61022, Україна
wdpgames@yandex.ru, demenjay@gmail.com, nlfsr01@gmail.com, vshlokin@ukr.net

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

Надійшло: Квітень 2019.

Анотація: *Обговорюється характеристики і особливості роботи в системі Hyperledger Fabric та характерні проблеми реалізації транзакцій. Проведено розгляд регістрів і принципів їх роботи в відповідних системах, зокрема підключення системи до голосування та перевірка блоків блокчейна. Визначено механізми захисту відповідних систем і характерні уразливості. Розглянуто інфраструктуру відкритого ключа в системі.*

Ключові слова: *комп'ютерні мережі; децентралізація; Hyperledger Fabric; блокчейн.*

1 Вступ

В представленій роботі розглядаються питання що пов'язані зі створенням системи голосування на основі блокчейн проекту - Hyperledger Fabric, яка, свого часу, була створена для побудови бізнес додатків корпоративного рівня [1], як проект в рамках IBM. Зі слів представників IBM, Hyperledger Fabric «розроблена для створення основи побудови мережевих блоків в корпоративному класі, які можуть швидко масштабуватися, оскільки нові учасники мережі приєднуються та здійснюють транзакції зі швидкістю більше 1000 транзакцій в секунду в великих екосистемах». Ця платформа (надалі Hyperledger) має відкритий вихідний код, надає деякі ключові можливості для диференціації порівняно з іншими популярними платформами.

Hyperledger є блокчейн системою з цифровим регістром, який виявляє спроби підробки та є захищеними від несанкціонованого втручання. Такі системи зазвичай реалізовані без центрального сховища та без централізованого управління. На своєму базовому рівні вони дозволяють співтовариству її користувачів реєструвати транзакції в загальному реєстрі в рамках цієї спільноти, таким чином, що при штатному функціонуванні блокчейн мережі, ніяка транзакція (теоретично) не може бути змінена після її опублікування. В загальному випадку блокчейн - це розподілені цифрові регістри транзакцій які криптографічно підписані, та згруповані в відповідні блоки. Кожен з них криптографічно пов'язаний з попереднім блоком після проведення відповідної перевірки та прийняття консенсусного рішення. Згодом, з додаванням нових блоків, старі блоки починають важко піддаватися змінам (*тобто створюється опір підробки*). Нові блоки записуються в копії реєстру блокчейн мережі, а будь-які конфлікти вирішуються автоматично, відповідно до встановлених правил [2].

Одним з ключових моментів диференціації є те, що Hyperledger був створений саме під Linux Foundation, який має успішну історію створення та просування проектів з відкритим кодом під управлінням яких, як показала історія, зростають сильні спільноти та процвітають екосистеми. Як стверджує спільнота розробників програмного забезпечення - відкритий вихідний код є основною «філософією» цього проекту: - «Тільки відкритий підхід до розробки програмного забезпечення може забезпечити прозорість, довговічність і підтримку, необхідних для просування технологій блокчейн в майбутньому» [3].

Одним з найважливіших диференціаторів платформи є підтримка підключення консенсусних протоколів, які дозволяють платформі більш ефективно налаштовуватися відповідно до конкретних випадків використання та моделей довіри.

Hyperledger може використовувати консенсус-протоколи, які не вимагають нативної крипто валюти для дорогого видобутку або для розробки сматр-контракту. Уникнення криптовалюти зменшує деякі значні вектори ризику/атаки, а відсутність криптографічних видобувних операцій означає, що платформа може бути розгорнута приблизно з тими ж операційними витратами, як і будь-яка інша розподілена система.

В основному цим проектом зацікавлені в таких галузях, як охорону здоров'я, послуги кредитних карток (фінансовий сектор), ланцюг постачання продукції та виробництво.

Розробники Hyperledger на своєму сайті навели п'ять основних цілей свого проекту:

- Забезпечення нейтральної, відкритої та керованою громадою інфраструктури, що підтримується технічним та бізнес-управлінням;
- Створювання корпоративних класів, відкритих джерел;
- Створення технічних спільнот для розробки блокчейну;
- Ознайомлення громадськості з можливостями блокчейн технологій;
- Заохочування спільнот, використовуючи підхід з багатьма платформами.

Характеризуючи архітектуру Hyperledger, можна виділити такі аспекти, як:

- Підтримка підприємств, яка може забезпечити ступінь стабільності, мотивуючи тих, хто ще не впевнений в перспективах блокчейн;
- Модульна архітектура, в котрій саме користувач визначає, що йому використовувати, а що ні.
- Розподілена книга Hyperledger та платформа смарт-контрактів дозволяють використовувати приватні канали.

Наприклад: - *якщо у вас є велика блокчейн мережа і ви хочете поділитися окремими даними лише з певними сторонами, то ви можете створити приватний канал лише с цими учасниками.*

- Прозорий процес. Самі транзакції можуть бути і не прозорими, але сам процес розробки – може. «На цьому етапі головні команди Hyperledger були надзвичайно готові збалансувати потреби, щоб отримати важливі віхи з відкритим і прозорим процесом розвитку» - зазначив засновник системи «Skuchain» - Закі Манян.
- Смарт-контракти: - подібно до Ethereum, Hyperledger дозволяє використовувати смарт-контракти, які називаються "*ланцюговими кодами*".

Для написання смарт-контрактів (*чейнкод в контексті Hyperledger*) використовують Golang (хоча Hyperledger дозволяє використовувати і інші мови). А для розробки користувальницького додатка використовувався Node.js, Java та Go з відповідним Hyperledger Fabric SDK, а не обмежені доменні мови (DSL). Це означає, що більшість підприємств вже мають набір навичок, необхідних для розробки інтелектуальних контрактів, і тому не потрібно додаткового навчання для вивчення нової мови або DSL [4].

На відміну від Ethereum, Fabric не вимагає вбудованої криптовалюти. Можна розробити національну валюту або цифровий маркер з ланцюговим кодом, але це вимагатиме значного інвестування ресурсів розвитку. Тому відсутність криптовалют робить його більш практичним для створення бізнес–додатків корпоративного рівня.

Серед недоліків слід зазначити проблему масштабування [5]. Так як Hyperledger побудований на концепції каналів, то це значить, що для кожної зі сторін, які виконують певні операції, буде потрібно створити свій канал. Вочевидь, це може призвести до створення великої кількості каналів, та як наслідок, система вийде з під контролю. Нижче розглянемо питання щодо створення та керування транзакцій блокчейна та їх механізми захисту на основі реалізації системи голосування.

2 Регістр в системі Hyperledger

Регістр в системі Hyperledger являє собою збори транзакцій. Протягом всієї історії рукописні та паперові бухгалтерські книги використовувалися для відстеження обміну товарів і послуг. У наш час регістри зберігаються в цифровому вигляді, більшою частиною у великих

базах даних, що належать і управляються централізованою довіреною третьою стороною (*власником регістру*) від імені спільноти користувачів. Ці регістри з централізованою власністю можуть бути реалізовані *централізованим* або *розподіленим* способом (або єдиний сервер або координується кластер серверів).

В даний час все більше зростає інтерес до вивчення можливості розподілу власності на регістр. Технологія блокчейн дозволяє реалізувати саме такий підхід, використовуючи при цьому як розподілене володіння, так і розподілену фізичну архітектуру.

Розподілена фізична архітектура мереж блокчейн часто включає в себе значно більшу кількість комп'ютерів, ніж це характерно для розподіленої фізичної архітектури з централізованим управлінням. Зростаючий інтерес до розподіленої власності на регістри пояснюється можливими проблемами з довірою, безпекою і надійністю, що пов'язані з регістрами та централізованою власністю. Деякі важливі особливості цих двох систем наведені нижче:

1. Регістри з централізованим видом володіння можуть бути втрачені або знищені, а користувач, при цьому, повинен цілком покладатися на те, що власник підтримує систему належним чином.

• Мережа блокчейн є розподіленою за своєю суттю. Вона створює безліч резервних копій, кожна з яких постійно оновлюється і синхронізується з одними і тими ж обліковими даними між всіма одноранговими регістрами. При цьому ключовою перевагою технології блокчейн є те, що кожен користувач може зберегти свою власну копію реєстру, що значно ускладнює втрату або руйнування регістра. Слід звернути увагу на те, що певні реалізації мережі блокчейн забезпечують можливість проведення закритих транзакцій або закритих каналів. Принциповим є те, що закриті транзакції сприяють доставці інформації тільки тим сайтам, які беруть участь в транзакції, а не всієї мережі.

2. Регістри з централізованим володінням можуть перебувати в однорідної мережі, де все програмне та апаратне забезпечення і мережева інфраструктура може бути однаковими. В цих умовах, атака на одну частину мережі буде працювати на всіх її елементах, тому загальна стійкість системи може бути значно знижена.

• Блокчейн - це гетерогенна структура, де у силу великої кількості всіляких відмінностей між її вузлами, атака на один них не завжди буде працювати на інших її частинах.

3. Регістри з централізованим володінням, частіше за все, розташовані повністю в певній географічній локації (наприклад, все в одній країні). Тому в разі виникнення будь яких перебоїв в роботі мережі, регістр і служби, скоріш за все будуть недоступні.

• Мережа блокчейн, зазвичай, складається з географічно розподілених вузлів, що знаходяться в будь-якій точці світу. Внаслідок цього, а також рівноправній моделі організації роботи мережі блокчейн, вона більш стійка до втрати одного, або навіть цілої ділянки вузлів.

4. Транзакції в регістрі з централізованим володінням не проводяться прозоро і можуть бути недійсними. В цих умовах користувач може тільки вірити, що власник перевіряє всі отримані транзакції.

• Мережа блокчейн зобов'язана перевіряти дійсність всіх транзакцій. Якщо шкідливий вузол передає недійсні транзакції, то інші будуть їх виявляти і ігнорувати, таким чином запобігаючи поширення недійсних транзакцій по всій блокчейн мережі.

5. Список транзакцій в регістрі системи з централізованим управлінням може бути не повним, а її користувач повинен вірити, що власник записує всі отримані транзакції.

• Мережа блокчейн утримує всі прийняті транзакції всередині розподіленого регістру. При цьому, для того щоб побудувати новий блок, необхідно послатися на попередній. Отже, надбудувати нового над старим блоком. Таким чином, якщо вузол не містить посилання на останній блок, то інші вузли повинні його відкинути.

6. Інформація об операції в регістрах з централізованим управлінням може бути піддана змінам. Користувач повинен вірити, що власник не змінює минулі транзакції.

• Мережа блокчейн з метою забезпечення виявлення підробок та захисту від зовнішнього втручання в регістри, використовує відповідні криптографічні механізми (*цифровий підпис і криптографічна хеш-функція*).

7. Системи з централізованим володінням можуть бути небезпечними (*з точки зору можливостей парирування актуальних загроз безпеки*). Користувач повинен був упевнений, що пов'язані комп'ютерні системи та мережі своєчасно отримують критичні оновлення для операційних систем і систем інформаційної безпеки та впроваджують передові методи захисту. Така система може бути зруйнована, а чутлива інформація може бути вкрадена або скомпрометована.

• У мережі блокчейн, в силу її розподіленої структури, не може бути централізованої точки атаки. В цілому, інформація в блокчейн мережі загальнодоступна і, відповідно, нічого красти. В цих умовах щоб зробити атаку на користувачів блокчейн мережі, атакуючий повинен особисто визначити кожну з цілей. Вибір в якості мети всієї мережі блокчейн, зустріне опір справжніх вузлів, представлених у системі. Якщо один вузол був змінений, то це вплине тільки на нього, але не на систему в цілому.

Для прикладу розглянемо модель системи голосування, що була попередньо розроблена для ознайомлення з інфраструктурою Hyperledger та її тестування. Ця система має щонайменше одного адміністратора, готовий механізм взаємодії учасників та можливість породжувати і контролювати нові вузли. Це означає, що можна вільно додавати учасників, активи та проводити транзакції [6].

2.1 Підключення системи голосування до Hyperledger і перевірка блоків блокчейна

Для підключення моделі голосування треба ввести команду *composer network install --card PeerAdmin@hlfv1 --archiveFile voting-simple@0.0.2.bna* в термінал, щоб встановити зв'язок з системою голосування. Команда *./createPeerAdminCard.sh,* дозволяє створити карту доступу для отримання прав в участі для голосування.

Тест працездатності мережі забезпечується командою *composer network ping --card admin@voting-simple*. За допомогою команди *composer-rest-server*, згенеруємо REST-сервер.

Далі йде серія питань, які будуть визначати параметри майбутнього сервера. Відповіді вводяться з клавіатури та підтверджуються натисканням клавіші «Enter». Для забезпечення коректної роботи моделі слід ввести параметри, що уточнюють як повинна працювати система, або за допомогою команди *composer-rest-server -c admin@voting-simple -n never -u true -w true* пропустимо етап для визначення параметрів сервера. Після цього у вікні терміналу з'явиться повідомлення:

Web server listening at: http://localhost:3000

Browse your REST API at http://localhost:3000/explorer.

Його відображення означає, що сервер працює.

Ознайомитися зі структурою системи можна за адресою *http://localhost:3000/explorer***.**

Після запуску сервера введемо в новому вікні терміналу наступну команду *docker ps –a*, що дозволить побачити інформацію про запущені контейнери вузли мережі.

Потрібно вибрати CONTAINER ID порту hyperledger / fabric-peer-1.2.1 і наступним кроком ввести в консоль команду *docker exec-it CONTAINER ID/bin /bash* (перейдемо в кореневе розташування цього контейнера).

Наступним шагом перейдемо в іншу директорію за допомогою команди *cd /var/hyperledger/production/ledgersData/chains/chains/composerchannel/* команда *ls* **(**виводить на консоль файли, що знаходяться у поточній директорії).

Для виведення блоку блокчейна використовується команда *peer chaincode query -C "composerchannel" -n qscc –c '{"Args":["GetBlockByNumber","composerchannel","Number"]}'*, замість Number термінал вводиться номер блоку (для виводу на екран). Блок складається з нуля і з кожним голосом регістр блоку перезаписує свою транзакцію, створюючи новий блок зі старими операціями залишаючи, при цьому, історію в попередньому блоці.

Відображення голосу кандидата в регістрі блокчейн виглядає наступним чином:

```
[
  {
        "$class": "org.voting.example.vote", - модель системи голосування;
```

"candidateVoteAsset": {Alice}, - ім'я кандидата;
"ifVotedAsset": {false/true}, - перевірка що кандидат проголосував;
"transactionId":" string", - транзакція записується в рядок;
"timestamp": "2019-05-06T22:24:06.804 Z" - час коли голос був доданий.
}
]

3 Захист та ключі в системі Hyperledger

Щоб зрозуміти систему захисту особистих даних, слід уявити ситуацію, в якої деяка особа відвідуєте супермаркет, щоб щось купити продукти, але на касі присутній знак, який оголошує, що приймаються тільки картки Visa, Mastercard і AMEX. В цьому разі, якщо особа буде намагатися розплатитися іншою карткою - назвемо її "ImagineCard" – буде зовсім неважливо, чи є ця карта справжньою та чи є на цьому рахунку достатньо коштів - в будь-якому випадку це буде неможливо виконати. В цих обставинах мати дійсну кредитну картку явно недостатньо – вона, також, повинна бути прийнята в конкретному магазині!

PKI та MSP (англ. *Membership Service Provider* - постачальник послуг членства) працюють разом однаково - PKI надає список ідентичностей, а MSP говорить, хто з них є членами даної організації, яка бере участь у мережі.

Органами сертифікації PKI та MSP надаються подібні комбінації функціональних можливостей. PKI подібний до постачальника карт - він видає велику кількість різних типів перевірених облікових записів. MSP, з іншого боку, подібний до списку постачальників карток, прийнятих магазином, визначаючи, які ідентичності є довіреними членами (*учасниками*) платіжної мережі магазину. MSP перетворюють перевірені ідентичності на членів мережі блокчейн.

Інфраструктура відкритого ключа (PKI – від англ. *Public Key Infrastructure*) - це сукупність інтернет-технологій, що забезпечують безпечні комунікації в мережі**.** Фактично це PKI, який додає S в HTTPS.

Відомі чотири ключові елементи для PKI: - цифрові сертифікати; - відкриті та приватні ключі; - органи сертифікації; - списки анулювання сертифікатів.

Цифровий сертифікат - це документ, який містить набір атрибутів, що стосуються власника сертифіката. Найбільш поширеним типом сертифікату є той, що відповідає стандарту X.509. Він дозволяє кодувати ідентифікаційні дані сторони в його структурі.

Сертифікати можуть бути широко розповсюджені, оскільки вони не включають в себе особистих ключів учасників та цифровий підпис. Вони можуть бути використані як інструмент довіри для автентифікації повідомлень, що надходять від різних учасників.

Центр сертифікації (ЦС), також має сертифікат, що є в широкому доступі. Це дозволяє споживачам ідентифікаційних даних, що видані даним ЦС, перевіряти їх, маючи на увазі що сертифікат може бути створений тільки власником відповідного закритого ключа.

У налаштуваннях блокчейн, кожен користувач, що бажає взаємодіяти з мережею, потребує забезпечення його ідентичності. У цьому разі ЦС забезпечує основу того, щоб всі користувачі організації мали перевірену цифрову ідентичність.

Традиційними механізмами автентифікації є цифрові підписи, які забезпечують гарантії цілісності підписаного повідомлення. З технічної точку зору, механізми цифрового підпису вимагають, щоб сторона мала два ключа – відкритий та приватний (останній служить для створення цифрових підписів на повідомленнях). Взаємозв'язок між цими ключами робить можливим захищені комунікації. При цьому математична залежність між обома ключами така, що закритий ключ може використовуватися для створення підпису на повідомленні, яке може відповідати тільки відповідний відкритий ключ, і тільки на одне і те ж повідомлення.

Приклад використання приватного ключа [7] для підписання умовного повідомлення наведено на рис. 1. Згідно з ним Мері використовує свій приватний ключ для підписання повідомлення. При цьому підпис може перевірятися всіма, хто бачить підписане повідом-

лення за допомогою відкритого ключа. Цифрові підписи створюються за допомогою ECDSA. Слід підкреслити, що найважливішою перевагою ECDSA є можливість його роботи на значно менших полях . Як, загалом, з криптографією еліптичної кривої, передбачається, що бітовий розмір відкритого ключа, який буде необхідний для ECDSA, дорівнює подвійному розміру секретного ключа в бітах.

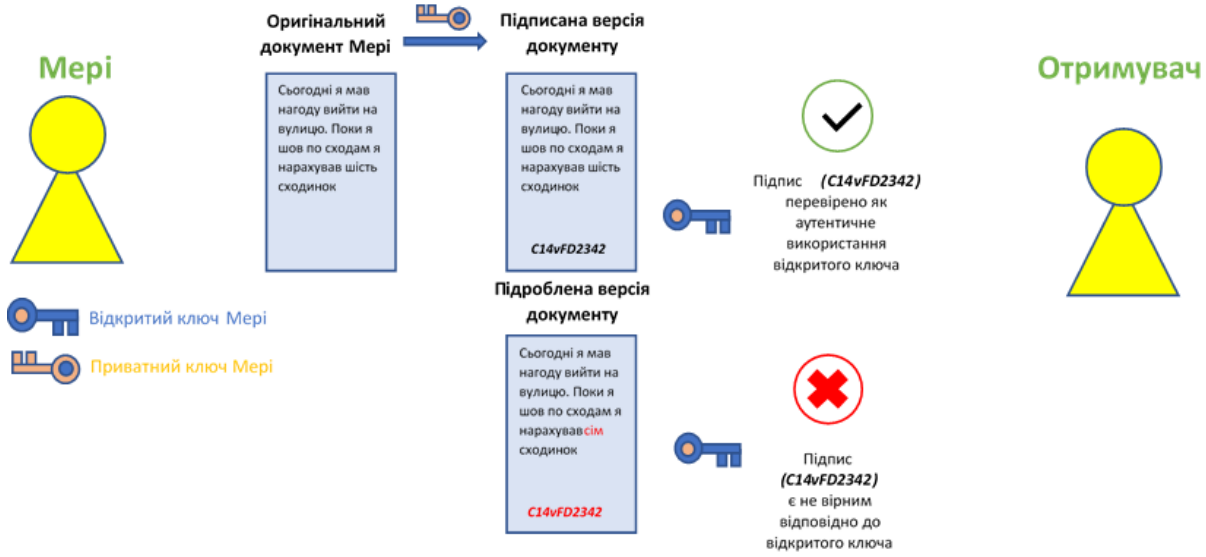


Рис. 1 – Використання приватного ключа для підпису повідомлення

У більшості випадків користувач зберігає облікові дані в себе, включаючи закритий ключ та він сам підписує транзакції. Проте варто зазначити, що деякі бізнес-сценарії можуть вимагати більш високого рівня конфіденційності. Однак слід зазначити, що в системі присутня проблема ключового зіткнення [8], але ця проблема не є специфічною для Hyperledger, це загальна проблема комп'ютерних наук і для існує багато способів її вирішення.

Повинно бути розуміння, що час коли SDK відправляє перший запит і час, коли ця транзакція зберігається в блокчейні, не є детермінованими. Це може зайняти кілька секунд, а більшість цих кроків виконуються паралельно. Отже, в чому тут проблема?

Розглянемо наступний приклад: є ланцюговий код, який переказує гроші від абонента Б до абонента А. Ланцюговий код повинен перевірити, що Б вистачає коштів до переказу, а якщо ні, то ця транзакція повинна бути скасована. Нехай на даний момент Б має 10 жетонів і поставив запит на передачу в бік А 10 з них. Чейнкод перевірить правильність цієї суми (*під час моделювання*), і ця транзакція буде надіслана замовнику для фіксації. Але, що станеться, якщо в той же час абонент Б послав би ще одну угоду, але вже до абонента В, і знову ж таки на 10 жетонів?

В цьому разі чейнкод перевірятиме, чи має він (абонент Б) необхідну суму, оскільки передача до абонента А буде перевіряється (*в симуляції*), але не здійснюється. Результатом буде те, що абонент Б відправляє 20 жетонів, у той час як у нього фактично є тільки 10. Це є класичний приклад "подвійної проблеми витрат".

З цієї причини Hyperledger реалізує MCVV - добре відомий і перевірений механізм для запобігання подібним ситуаціям. В цілому Hyperledger не дозволить користувачам створити такий тип ситуації - конфліктні транзакції будуть відхилені, а SDK буде проінформований про відповідну відмову.

В чому полягає проблема цієї архітектури? Якщо хтось намагаєтеся оновити один і той же ключ, всі транзакції, крім одного, будуть відхилені. Так наприклад, якщо, у вас є IoT датчик (від англ..*internet of things* - Інтернет речей), що передає дані з частотою 10 разів на секунду, а середній час створення нового блоку (для внесення даних) становить 1 сек, то 9 з 10 транзакцій будуть відхилені, оскільки вони конфліктують. Але уявимо

собі такий же сценарій з грошовим переказом: - це приклад дуже поганого досвіду так, як 9 з 10 фінансових транзакцій не вдасться.

В цьому контексті не варто забувати, що Hyperledger може працювати з частотою більш ніж 210 000 транзакцій за хвилину. Таким чином при проектуванні архітектури, структур даних і потоків необхідно враховувати можливості MVCC Hyperledger, а методи, які можуть бути використані, визначаються типами даних і потоками. Немає жодного «золотого правила» або «кожного з них».

Варто зазначити, що MVCC розшифровується, як керування паралельним доступом за допомогою багатоверсійності [9]. Без управління паралельного доступу, якщо хтось одночасно зчитує та записує інформацію бази даних, є можливість, що читач побачить тільки наполовину написану або непослідовну інформацію.

Ізоляція – це властивість, що гарантує паралельний доступ до даних. Ізоляція реалізується за допомогою протоколу управління паралельним доступом. Найпростішим методом буде змусити всіх читачів чекати поки записувач не закінчить свої дії, що відомо як *блокування читання-запису*. Я показав досвід такі блокування часто викликають суперечки, особливо між довгими транзакціями читання та обновлення. Задум MVCC полягає в тому, щоб зберігати декілька копій кожного елементу даних. Таким чином, кожен користувач, що під'єднаний до бази даних в конкретний період часу бачить лише так званий «знімок» бази даних. При цьому будь-які зміни, що здійснюються записувачем не будуть відображатися для інших користувачів до моменту безпосереднього завершення відповідних змін. Тому архітектура, структури і потік повинні бути адаптовані таким чином, щоб не відбувалося жодних конфліктуючих операцій або щонайменше їх виконувалось дуже мало.

Бази даних, які використовують MVCC, є центральними, вони виконують більшість операцій в пам'яті, можуть використовувати дуже низький рівень блокування і синхронізації процесу, а більшість операцій займають мілісекунди. Так як Hyperledger є децентралізованою, то багато її учасників можуть бути або в автономному режимі, або мати величезне мережеве відставання. Також в чейнкоді Hyperledger немає дубльованих ключів. Оскільки зіткнення відбувається тільки тоді, коли один і той же ключ оновлюється одночасно, а якщо ніколи не використовуємо той же ключ, то зіткнення ніколи не відбудеться. Це відносно просто реалізувати в ланцюговому коді, але пізніше викличе багато інших питань. Замість того, щоб мати один ключ для облікового запису, користувач буде мати, можливо, сотню ключів, і знати фактичну суму в цьому коді ланцюжка рахунку повинні приймати всі ці ключі, перебирати їх і визначати те, що є поточним значенням. Слід розуміти, що в цьому разі ви/користувач "обходите" MVCC, та відкриті для подвійної проблеми з витратами. Це можливо, тому що об'єктивне значення, яке перевіряє MVCC, не є одним ключем, а це є результат іншого процесу - ітерація всіх ключів і виконання певних математичних дій над значеннями. Цей підхід корисний, коли потрібно зберігати величезну кількість швидких даних, що зберігаються незмінно в блокчейні, але ці дані не будуть частиною бізнес-процесу.

Так, щоб отримати файл с ключами треба спочатку отримати відповідний CONTAINER ID за допомогою команди *docker ps*. Контейнер, котрий потрібний буде стояти під командою *"peer node start"* та буде мати образ *hyperledger/fabric-peer:1.2.1*.

Щоб отримати bash-оболонку контейнера Hyperledger використовують команду *docker exec /bin/bash*. Зазвичай, для того, щоб взнати, що міститься в команді використовують команду *docker exec -it <container name> <command>*, в нашому випадку це є доступ до корінної директорію контейнеру. *cd /var/hyperledger/production/* - команда, що веде в ще одну внутрішню директорію, а *cd /etc/hyperledger/peer/msp/keystore/* буде шляхом до нашого файлу, який знаходиться в цій папці.

ЦС є настільки важливими, що Fabric надає вбудований компонент ЦС, якій дозволяє створювати CA в мережах блокчейн, котрий ви формуєте. Fabric ЦС - приватний кореневий постачальник ЦС, здатний керувати цифровими ідентифікаціями учасників Fabric, які мають форму сертифікатів X.509. Оскільки Fabric ЦС є користувацьким ЦС, що орієнту-

ється на потреби кореневого ЦС у Fabric, то воно не здатне забезпечити SSL-сертифікати для загального/автоматичного використання в браузерах.

Крім того, для посилення рівня безпеки в системі використовуються відповідні списки анулювання сертифікатів (CRL – *Certificate Revocation List*). По суті це список посилань на сертифікати, про які ЦС знає що вони відкликанні. Так наприклад, відносно умов/сценарію магазину, CRL буде схожий на список викрадених кредитних карт.

В умовах коли третя сторона хоче перевірити особистість іншої сторони, вона спочатку перевіряє CRL відповідного ЦС, щоб переконатися, що сертифікат не був відкликаний. В цьому разі верифікатору не потрібно перевіряти CRL, але якщо вони цього не роблять, то ризикують прийняти порушений ідентифікатор. Слід звернути увагу на те, що відкликаний сертифікат сильно відрізняється від сертифіката, який закінчив свій термін дії. Термін дії анульованих сертифікатів не закінчився - вони, за будь-яким іншим показником, є повністю дійсним сертифікатом.

PKI може надавати перевірочні ідентичності через ланцюжок довіри. Наступним кроком є те, як ці ідентичності можна використовувати для представлення надійних членів мережі блокчейн. Саме там вступає в дію постачальник послуг членства (MSP) - він визначає сторони, які є членами даної організації в блокчейн мережі.

4 Висновки

Технологія блокчейн ґрунтується на розподіленому зберіганні частки інформації на кожному з вузлів, що підключені до системи, а кожен блок містить посилання на інші частини бази даних. Внесення змін відбувається послідовно, шляхом перезапису кожного блоку, а допуск до відповідних дій здійснюється за допомогою індивідуального логіна і пароля.

Основні переваги систем що впроваджують блокчейн полягають в наступному:

1. Безпека. Зламати систему складно - необхідно отримати доступ до механізму консенсусу який залежить від кількості вузлів в децентралізованій системі.
2. Відсутність комісійних зборів. Транзакції здійснюються безпосередньо між користувачами.
3. Висока швидкість роботи. Блоковий підхід використовує продуктивність всіх учасників/вузлів, розділяючи між ними практично всі навантаження.
4. Безперервність роботи. Блокчейн система діє 24 години на добу, та 365 днів на рік.

Блокчейн технологію вже сьогодні успішно застосовують не тільки для фінансових транзакцій, але і для зберігання інформація, а в перспективі можливе використання блокчейна в сферах освіти, комунальних службах та промисловості.

Однак блочне зберігання, не є ідеальними. До його мінусів слід віднести наступне:

1. Відсутність державного регулювання. На законодавчому рівні взаємовідносини в децентралізованих мережах поки ніяк не регламентуються.
2. Анонімність приваблює злочинні елементи, так як при здійсненні транзакцій не потрібно підтверджувати особу.
3. Відносна новизна технології викликає певну недовіру з боку великих компаній і корпорацій. Однак, стрімкий розвиток та збільшення популярності відповідних систем поступово переконує про перспективність проектів на основі блочного зберігання інформації.
4. Використання технології блокчейн все ще знаходиться на початковій фазі розвитку, але воно впроваджує зрозумілі та ґрунтовні криптографічні принципи.
5. Блокчейн використовує існуючі мережеві і криптографічні технології, однак, використання існуючої інфраструктури здійснюється на принципово нових принципах.

Таким чином технологія блокчейн - це новий перспективний інструмент з потенційними додатками для організацій, що здатен забезпечити високій рівень безпеки транзакцій без необхідності існування вузла централізованого управління мережею.

Всі транзакції захищені криптографічними хешами, підписуються і перевіряються з використанням пар асиметричних ключів. Історія транзакцій ефективно і безпечно фіксує вісь ла-

нцюжок подій таким чином, що будь-яка спроба підміни або будь якої корекції минулої транзакції, вимагатиме перерахунку всіх наступних блоків транзакцій.

Посилання

[1] Hyperledger Fabric 2019. Instroduction URL: https://hyperledger-fabric.readthedocs.io/en/master/whatis.html.
[2] NISTIR 8202 Blockchain Technology Overview / Yaga D., Mell P., Roby N., Scarfone K. URL: https://csrc.nist.gov/ publications/detail/nistir/8202/final
[3] Нефедов Н. Hyperladger Fabric для чайников. URL: https://habr.com/ru/company/ibm/blog/444874.
[4] The plus and cons of hyperledgerfabric. URL: https://www.verypossible.com/blog/the-pros-and-cons-of-hyperledger-fabric.
[5] Установка інструментів Hyperledger Fabric для розробки та тестування блокчейн-мереж. URL:https://docs.google.com/document/d/1NWVvRCIHphirDHD169AYgT2VG5b4xhuFYC8N9WdF3Cw/edit#heading=h.f3a2riitnib6.
[6] Head_aefkz. Плюсы и минусы блокчейна. URL: https://aef.kz/blockchain/plyusy-i-minusy-blokchejna
[7] Hyperledger Fabric 2019. Identity. URL: https://hyperledger-fabric.readthedocs.io/en/master/identity/identity.html
[8] Refs and Transaction. URL: https://clojure.org/reference/refs
[9] Vankov I. How to prevent key collisions in Hyperledger Fabric chaincode. URL: https://medium.com/@gatakka/how-to-prevent-key-collisions-in-hyperledger-fabric-chaincode-303700716733.

Reviewer: Serhii Kavun, Doctor of Sciences (Economics), Ph.D. (Engineering), Full Prof., Kharkiv University of Technology "STEP", Malom'yasnitska St. 9/11, Kharkiv, 61010, Ukraine. E-mail: kavserg@gmail.com

Received on Aprilr 2019.

Authors:
Nikita Goncharov, computer science student, V.N. Karazin Kharkiv National University, Ukraine.
E-mail: wdpgames@yandex.ru

Eugene Demenko, computer science student, V.N. Karazin Kharkiv National University, Ukraine.
E-mail: demenjay@gmail.com

Nikolay Poluyanenko, Ph.D., Computer Science Department, V.N. Karazin Kharkiv National University, Ukraine.
E-mail: nlfsr01@gmail.com

Vladimir Shlokin, Director of the Innovation Center, V.N. Karazin Kharkiv National University, Ukraine.
E-mail: vshlokin@ukr.net

Building a voting system with using blockchain technologies in the example of Hyperledger.

Abstract. It discusses the characteristics and features of work in the Hyperledger Fabric system and the characteristic problems of the implementation of transactions. The registers and principles of their work in the respective systems are considered, in particular, the connection of the system to voting and the check of Blockchain blocks. Defined mechanisms for the protection of relevant systems and characteristic vulnerabilities. Considered public key infrastructure in the system.

Keywords: Computer networks; Decentralization; Hyperledger Fabric; Blockchain.

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

Поступила: Апрель 2019.

Авторы:
Никита Гончаров, студент факультета компьютерных наук, Харьковский национальный университет имени В. Н. Каразина, Харьков, Украина. E-mail: wdpgames@yandex.ru

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

Николай Полуяненко, к.т.н., викладач факультета компьютерных наук, Харьковский национальный университет имени В. Н. Каразина, Харьков, Украина. E-mail: nlfsr01@gmail.com

Владимир Шлокин, директор инновационного центра, Харьковский национальный университет имени В. Н. Каразина, Харьков, Украина. E-mail: vshlokin@ukr.net

Построение системы голосования с использованием блокчейн технологий на примере Hyperledger.

Аннотация. Приведена теоретическая информация о тестировании программного обеспечения методом фаззинга. Рассмотрены технологии обучения с подкреплением и интеллектуального фаззинга в процессе тестирования программного обеспечения. Описан алгоритм, с помощью которого реализуются указанные методы и технологии. Предложены статистические результаты исследований, которые были проведены во время тестирования некоторых программ и утилит, предназначенных для повседневного использования, а также программы разработанной студентами.

Ключевые слова: компьютерные сети; децентрализация; Hyperledger Fabric; блокчейн.

УДК 004.94: 004.7.056 DOI: 10.26565/2519-2310-2019-1-06

ДИСПЕРСИОННЫЙ АНАЛИЗ СЕТЕВОГО ТРАФИКА ДЛЯ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ В SMART GRIDS

Александр Кузнецов, Влада Григоренко, Андрей Дьяченко, Михаил Багмут

Харьковский национальный университет имени В.Н. Каразина, Харьков, 61022, Украина
kuznetsov@karazin.ua, mrsgriggy@gmail.com, andrey.090220@gmail.com, sapsanmiha@gmail.com

Рецензент: Владимир Максимович, д.т.н, проф., Институт компьютерных технологий, автоматики и метрологии Национального университета «Львовская политехника», Львов, 79013, Украина.
vmax@polynet.lviv.ua

Поступила: Апрель 2019.

Аннотация: *Рассматриваются системы обнаружения и предотвращения вторжений в современных телекоммуникационных системах и сетях. Исследуются методы мониторинга событий, состоящие в анализе сетевой активности отдельных служб и информационных сервисов. Предлагается использовать математический аппарат дисперсионного анализа для обработки результатов моделирования телекоммуникационных систем и исследования статистических свойств сетевого трафика при определении значимости расхождения или совпадения характеристик. Предлагаемый подход состоит в использовании статистического критерия Фишера, основанного на оценке отношения выборочных дисперсий. Это позволяет с заданным уровнем значимости проверять гипотезу об однородности статистических свойств сетевого трафика относительно показателя рассеивания (дисперсии). Полученные результаты экспериментальных исследований рекомендуется использовать для совершенствования механизмов мониторинга сетевой активности отдельных служб и информационных сервисов, в том числе и для обнаружения и предотвращения вторжений в телекоммуникационных системах и сетях* перспективных *Smart Grids.*

Ключевые слова: *телекоммуникационные системы и сети; системы защиты и предотвращений вторжений; дисперсионный анализ; кибербезопасность; Smart Grid.*

1 Вступ

Развитие информационных технологий содействует устойчивому росту количества пользователей различных электронных услуг и объемов обрабатываемых данных, повышению требований к скорости и надежности функционирования информационных, и компьютерных систем, а также появлению новых форм и способов реализации различных киберугроз. Особенно остро, в этом смысле, стоит вопрос обеспечения кибербезопасности современных систем энергоснабжения. Так, например, в результате лишь одной кибератаки на энергетическую систему Украины с использованием вредоносного кода типа BlackEnergy вечером 23 декабря 2015 года от сети электроснабжения было отключено 27 подстанций и 103 населенных пунктов. Другой пример нарушения функциональной безопасности связан с действием компьютерного вируса Petya.A (разновидность вируса WannaCry) летом 2017 года, в результате чего пострадали информационные системы национального банка Украины, «Укрпошти», «Укрзализниці», нескольких государственных и коммерческих банков, облэнерго, аэропортов, промышленных объектов и телерадиокомпаний. Таким образом, в условиях постоянной угрозы реализации различных кибератак, вопросы обеспечения кибербезопасности в современных информационно-коммуникационных системах чрезвычайно важны и тесно связанны с построением надежных и безопасных Smart Grids.

Проведенный анализ показал, что наибольшую уязвимость представляют методы сетевого управления, технологии доступа к электронным сервисам и услугам, а также процессы мониторинга состояния телекоммуникационных систем и сетей [1-13]. Под воздействием вредоносного программного кода отдельные коммуникационные и вычислительные компоненты могут быть несанкционированно переведены в нештатные режимы функционирования, приводящие к сбоям и нарушению установленного порядка их использования, уничтожению, искажению, блокированию, несанкционированной утечки обрабатываемой и передаваемой информации, а также к нарушению работы алгоритмов маршрутизации между узлами телекоммуникационных систем [2-4]. Очевидно, что разработка и исследование методов монито-

ринга сетевой активности, и совершенствование технологий обнаружения вредоносного программного кода в целях предотвращения несанкционированного воздействия на защищаемые инфокоммуникационные ресурсы, является актуальной научно-прикладной задачей. Качество решения этой задачи непосредственно влияет на обеспечиваемый уровень безопасности современных телекоммуникационных систем и применяемых информационных технологий.

В данной работе изучается возможность использования математического аппарата дисперсионного анализа для целей исследования свойств сетевого трафика различных служб и информационных сервисов при определении значимости расхождения или совпадения их характеристик. Представленные результаты могут быть использованы для совершенствования механизмов мониторинга сетевой активности, в том числе для целей обнаружения и предотвращения вторжений в телекоммуникационных системах и сетях перспективных Smart Grids.

2 Анализ исследований и публикаций

Для обеспечения безопасности в современных телекоммуникационных системах и сетях применяются различные организационно-технические мероприятия, наиболее эффективные из которых состоят в построении интегрированных систем обнаружения (Intrusion Detection System – IDS) и предотвращения (Intrusion Prevention System – IPS) вторжений [2-13]. В основе функционирования современных образцов IDS и IPS находятся процедуры сбора, накопления, анализа и обработки информации о событиях, связанных с безопасностью защищаемой телекоммуникационной системы/комплекса. На основе результатов анализа (мониторинга) сетевой активности отдельных служб и сервисов, осуществляется принятие решения о текущем состоянии ресурсов защищаемой системы с выявлением и противодействием возможному несанкционированному использованию имеющихся инфокоммуникационных ресурсов [2-6].

Под системой обнаружения вторжений (СОВ) следует понимать программное или аппаратное средство, предназначенное для выявления фактов неавторизованного доступа в компьютерную систему или сеть либо несанкционированного управления [2-4]. СОВ обеспечивают дополнительный уровень защиты компьютерных систем за счет обнаружения некоторых типов вредоносной активности, которая способна нарушить безопасность компьютерной системы. К такой активности относятся сетевые атаки на наиболее уязвимые сервисы, атаки, направленные на повышение привилегий, неавторизованный доступ к критическим ресурсам, а также действия вредоносного программного обеспечения (компьютерных вирусов, троянов и червей) [2].

Под системой предотвращения вторжений (СПВ) следует понимать программную или аппаратную систему сетевой и компьютерной безопасности, обеспечивающую возможность обнаружения вторжений или нарушения установленных правил информационной безопасности и реализующую автоматическую защиту от выявленных нарушений [2-4].

Системы IPS следует рассматривать как расширение систем IDS с возможностью быстрого реагирования, путем реализации соответствующих действий по предотвращению выявленных атак или несанкционированных действий. Возможные меры предотвращения атак состоят в блокировке потоков трафика в телекоммуникационной сети, прерывании соединений, выдачи сигналов оператору и т.п. Кроме того IPS могут выполнять дефрагментацию и переупорядочивание пакетов TCP для защиты от пакетов с измененными SEQ (*номерами очереди*) и ACK (*номерами подтверждения*) [2].

Как правило, архитектура IDS включает в свой состав следующие элементы [2,3,6]:

- сенсорную подсистему (датчики), предназначенную для сбора событий, связанных с безопасностью защищаемой системы/комплекса;
- подсистему анализа, предназначенную для выявления атак и подозрительных действий на основе данных сенсоров;
- хранилище, обеспечивающее накопление первичных событий и результатов анализа;

- консоль управления, позволяющая конфигурировать IDS, отслеживать текущее состояние защищаемой системы и самой IDS, а также проводить аудит выявленных инцидентов.

В сетевой IDS (NIDS) сенсоры размещаются в наиболее важных точках сети (узлах сегментации), часто в демилитаризованной зоне, и на границе внешнего периметра сети [2,3,6,9]. Ее сенсоры прослушивают сетевой трафик в соответствующих точках и анализируют содержимое каждого пакета на присутствие вредоносных или не декларированных элементов. NIDS имеет доступ к сетевому трафику подключаясь к коммутаторам сети и отслеживает факты вторжения, проверяя сетевой трафик, ведя наблюдение сразу за несколькими узлами (хостами). Для предотвращения возможных инцидентов в состав NIDS входят соответствующее программные модули (блэйды), реализующие различные стратегии и способы парирования неавторизованных проникновений или атак защищаемых сетевых ресурсов (аппаратных и информационных).

Протокольные IDS (Protocol-based IDS, PIDS) используются для отслеживания трафика, нарушающего предусмотренные правила определенных протоколов либо синтаксис языка (например, SQL) [2,3]. PIDS представляет собой систему (либо агента), которая отслеживает и анализирует коммуникационные протоколы со связанными системами или пользователями. Так, для веб-сервера подобная СОВ обычно ведет наблюдение за HTTP и HTTPS протоколами. В случае использования HTTPS PIDS должна быть настроена таким образом, чтобы просматривать HTTPS пакеты еще до их шифрования и отправки в сеть.

Основанная на прикладных протоколах IDS (Application Protocol-based IDS – APIDS) – это система (или агент), которая ведет наблюдение и анализ данных, передаваемых с использованием специфичных протоколов [2,3]. Например, на веб-сервере с SQL базой данных, соответствующая IDS будет отслеживать содержимое SQL команд, передаваемых на сервер.

В узловых/хостовых IDS (Host-based IDS – HIDS) сенсор обычно является программным агентом, который ведет наблюдение за активностью узла сети, на который он установлен [2,3,13]. Для отслеживания вторжений проводится анализ системных вызовов, приложений, модификаций файлов (исполняемых, файлов паролей, системных баз данных), состояния узла сети и прочих источников.

Гибридная IDS совмещает два и более подходов к реализации СОВ. При этом данные от соответствующих агентов на хостах защищаемой системы комбинируются с сетевой информацией для формирования наиболее полного аудиторского следа.

Таким образом, в пассивной IDS при обнаружении аномалии безопасности, информация о фиксируемом нарушении записывается в соответствующий log-файл (хранилище данных), а уведомления тревоги отправляются на консоль и/или администратору системы по предусмотренному каналу связи.

В активной IPS системе в ответ на фиксируемые нарушения реализуются предусмотренные защитные реакции: - сбрасывается соединение или в соответствующих сегментах сети изменяются правила работы межсетевых экранов (МСЭ) и т.п.. Причем, ответные действия могут проводиться как автоматически, так и по командам администратора системы безопасности.

С учетом специфики размещения и характера выполняемых действий IPS классифицируются как [1-13]:

- *сетевые IPS* (NIPS) - отслеживают трафик в сети и блокируют подозрительные потоки данных [2-9];
- *IPS для беспроводных сетей* (Wireless Intrusion Prevention Systems, WIPS) - анализирует сетевую активность в беспроводных сетях. В частности, обнаруживают неверно сконфигурированные точки доступа к сети, неверно сконфигурированные МСЭ в зонах взаимодействия беспроводного и проводного сетевых сегментов, атаки «человек посередине», спуфинг MAC-адресов и др. [2-10];
- *поведенческий анализ сети* (Network Behavior Analysis, NBA) анализирует сетевой трафик и идентифицирует нетипичные потоки, например DoS и DDoS атаки [2-9, 11-12];

- *IPS для отдельных узл*ов (HIPS) содержит пакет специализированных резидентных программ, обнаруживающих подозрительную активность на узле ее установки [2-9, 13].

Обзор соответствующих источников [2-13] и анализ специфики условий работы соответствующих защитных решений показал, что наиболее эффективными направлением обеспечения высокого уровня информационной безопасности, в современных телекоммуникационных системах, является интегрирование различных решений IDS и IPS. В основе функционирования подобных решений лежит комплексное использование результатов анализа сетевой активности и активная имплементация разнообразных мер противодействия актуальным угрозам [2-13]. Такой подход можно считать наиболее перспективным и сбалансированным с точки зрения обеспечения требуемых показателей безопасности [2-5]. Проведенный анализ позволяет утверждать, что в основе работы наиболее развитых СОВ и СПВ лежит использование статистических данных о циркулирующем сетевом трафике и параметрах работы сетевого оборудования. Исследование этой информации имеет важное значение как для теоретического обоснования методов обнаружения и предотвращения вторжений, так и для разработки практических рекомендаций по построению программных и аппаратных средств мониторинга сетевой активности отдельных служб и информационных сервисов.

3 Методика экспериментальных исследований

Современные методы имитационного моделирования предоставляют широкие возможности по накоплению различных результатов статистических испытаний и эффективно проводить соответствующую обработку полученных данных, в частности, выполнять сравнение случайных параметров исследуемого процесса с целью определения значимости расхождения или совпадения их характеристик [14,15]. Один из наиболее развитых методов такой обработки, основанный на оценке отношений выборочных дисперсий, позволяет подтвердить или опровергнуть статистическую гипотезу об однородности результатов моделирования по показателю рассеивания (*дисперсии*) [15]. В рамках данной работы предлагается использовать математический аппарат дисперсионного анализа для обработки результатов моделирования работы телекоммуникационных систем и проведения исследований свойств сетевого трафика для различных служб и информационных сервисов при определении значимости расхождения или совпадения их характеристик.

Введем следующие обозначения и определения [14,15]. Пусть в результате эксперимента с имитационной статистической моделью, состоящего из N наблюдений получено N значений $x_1, x_2, ..., x_N$ исследуемой случайной величины X. По этим данным необходимо дать описание случайной величины X, т.е. необходимо определить ее характеристики. В практике моделирования и обработки экспериментальных данных довольно часто необходимо решать задачу подтверждения или опровержения гипотезы о принадлежности двух или более выборок одной генеральной совокупности. При этом признаки, по которым проводится сравнительная оценка, часто не являются детерминированными и обладают рассеиванием. Наиболее распространенной мерой рассеивания, используемой в теории вероятностей и математической статистике, является дисперсия (от лат. dispersio – рассеяние). В статистическом понимании дисперсия:

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - x^*)^2 ,$$

есть среднее арифметическое квадратов отклонений величин x_i от их среднего арифметического $x^* = (x_1 + x_2 + ... + x_n)/n$. Т.е. другими словами, дисперсия есть мера отклонения от статистического среднего.

В сложных технических системах дисперсия характеризует важные конструкционные и технологические показатели. В этом смысле при проведении исследований различных параметров технических систем наиболее важной характеристикой получаемых сравнительных оценок, является именно дисперсия, т.к. она обладает наибольшей общностью и позволяет, помимо прочего, проверять гипотезу равенстве средних значений выборок.

Таким образом, дисперсионный анализ является одним из эффективных механизмов исследования сложных технических систем и процессов, как наиболее общий и часто применяемый на практике метод сравнения качеств различных объектов.

Современные приложения дисперсионного анализа охватывают широкий круг задач и трактуются обычно в терминах статистической теории выявления систематических различий между результатами непосредственных измерений, выполненных при тех или иных условиях. Если значения неизвестных постоянных $a_1, ..., a_n$ могут быть измерены с помощью различных методов или измерительных средств $M_1, ..., M_m$ и в каждом случае систематическая ошибка может зависеть как от выбранного метода, так и от неизвестного измеряемого значения a_i, то результаты измерений x_{ij} представляют собой суммы вида:

$$x_{ij} = a_i, + b_{ij} + d_{ij}, i = 1, 2, ..., n; j = 1, 2, ..., m,$$

где b_{ij} – систематическая ошибка, возникающая при измерении a_i по методу/средством M_j, а d_{ij} – случайная ошибка.

Такую модель принято называть двухфакторной схемой дисперсионного анализа [14,15], где первый фактор – измеряемая величина, а второй – собственно, метод измерения.

Дисперсии эмпирических распределений, соответствующих множествам случайных величин x_{ij}, $x_{ij} - x_{i*} - x_{*j} + x_{**}$, x_{i*} и x_{*j}, где [14, 15]:

$$x_{i*} = \frac{1}{m}\sum_j x_{ij}\ ,\ x_{j*} = \frac{1}{n}\sum_i x_{ij}\ ,\ x_{**} = \frac{1}{n}\sum_i x_{i*} = \frac{1}{m}\sum_j x_{j*}$$

выражаются следующими выражениями:

$$s^2 = \frac{1}{mn}\sum_i\sum_j (x_{ij} - x_{**})^2\ ,\ s_0^2 = \frac{1}{mn}\sum_i\sum_j (x_{ij} - x_{i*} - x_{j*} + x_{**})^2\ ,$$

$$s_1^2 = \frac{1}{n}\sum_i (x_{i*} - x_{**})^2\ ,\ s_2^2 = \frac{1}{m}\sum_j (x_{j*} - x_{**})^2\ .$$

Эти дисперсии удовлетворяют тождеству [14,15]: $s^2 = s^2_0 + s^2_1 + s^2_2$, которое и объясняет происхождение названия дисперсионного анализа. Так если величины систематических ошибок не зависят от метода измерений (*т. е. между методами измерений нет систематических расхождений*), то отношение s^2_2/s^2_0 близко к единице. Это свойство лежит в основе *критерия для статистического выявления систематических расхождений*: - если s^2_2/s^2_0 значимо отличается от единицы, то гипотеза об отсутствии систематических расхождений отвергается. Значимость отличия определяется в согласии с законом распределения вероятностей случайных ошибок измерений. В частности, если все измерения равноточные и случайные ошибки подчиняются нормальному распределению, то критические значения для отношения s^2_2/s^2_0 определяются с помощью таблиц так называемого *F*-распределения (распределения дисперсионного отношения или распределения Фишера) [14,15].

Изложенная схема позволяет лишь обнаружить наличие систематических расхождений и, вообще говоря, непригодна для их численной оценки с последующим исключением из результатов наблюдений. Эта цель может быть достигнута только при многократных измерениях (при повторных реализациях указанной схемы).

Таким образом, *суть дисперсионного анализа* состоит в проверке гипотезы о тождественности выборочных дисперсий одной и той же генеральной совокупности [14,15].

Пусть имеются две выборки $x_1, x_2, ..., x_{N1}$ и $y_1\ y_2, ..., y_{N2}$ объемом N_1 и N_2, соответственно, случайных величин X и Y, имеющих нормальное распределение. Дисперсия случайной величины, являясь суммой квадратов ошибок, имеет распределение χ^2 (*распределение Пирсона*). Задача сравнения дисперсий случайных величин X и Y сводится к проверке исходной гипотезы (нулевой гипотезы H_0) о принадлежности двух выборок одной и той же генеральной совокупности [14,15].

Для проверки гипотезы о равенстве дисперсий используют независимую функцию, вычислимую по данным эксперимента. Такой функцией является *функция Фишера* (распределение Фишера, или *F*-распределение), а ее значение определяется как [14]:

$$F = \frac{U / k_1}{V / k_2},$$

где:

- U и V случайные величины, имеющие распределение χ^2;
- k_1 и k_2 соответствующие степени свободы случайных величин U и V соответственно, $k_1 = N_1 - 1$, $k_2 = N_2 - 1$;
- N_1 и N_2 – количество испытаний (объемы выборок).

Другими словами, случайная величина $F = \sigma_1^2 / \sigma_2^2$ имеет F-распределение, где: σ_1^2 и σ_2^2 – несмещенные оценки дисперсий, а x^* и y^* – несмещенные оценки математических ожиданий, полученных из независимых выборок, взятых из нормальных совокупностей:

$$\sigma_1^2 = \frac{1}{N_1}\sum_{i=1}^{N_1}(x_i - x^*)^2, \quad \sigma_2^2 = \frac{1}{N_2}\sum_{i=1}^{N_2}(y_i - y^*)^2, \tag{1}$$

$$x^* = (x_1 + x_2 + ... + x_{N_1}) / N_1,$$

$$y^* = (y_1 + y_2 + ... + y_{N_2}) / N_2. \tag{2}$$

Для подтверждения или опровержения гипотезы об однородности исследуемых выборок необходимо выбрать уровень значимости q, численно равный вероятности *неприемлемых* отклонений от принятой гипотезы.

Вид функции плотности распределения Фишера приведен на рис. 1, где также обозначены области неприемлемых значений F.

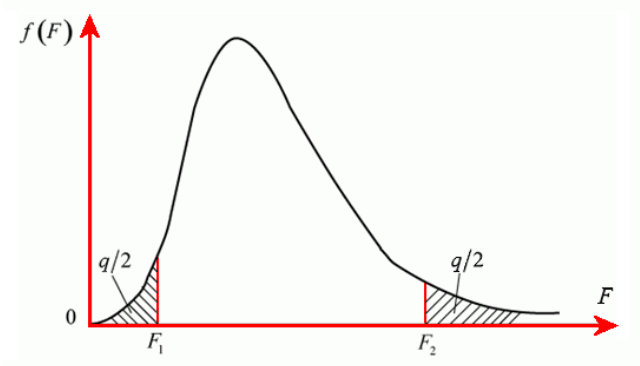


Рис. 1 - Плотность F-распределения

Граничные точки допустимых значений F определяются точками F_1 и F_2, соответствующих вероятностям $q/2$. Если вычисленное по данным эксперимента значение F попадает в область между граничными точками F_1 и F_2, т.е. не попадает в т.н. критическую область, принятая гипотеза не опровергается. *Чем меньше* уровень значимости q, *тем меньше* вероятность забраковать проверяемую гипотезу, когда она верна, т.е. совершить *ошибку первого рода*. Но *с уменьшением уровня значимости* (увеличения F_2) расширяется область допустимых ошибок, что приводит *к увеличению вероятности принятия неверного решения*, т. е. совершения *ошибки второго рода*. Следовательно, суждение о подтверждении или отклонении выдвинутой гипотезы высказывается **с** определенной степенью достоверности.

Задачу проводимых экспериментальных исследований сформулируем как задачу проверки гипотезы об однородности наблюдаемых трафиков различных телекоммуникационных служб и информационных сервисов по выборочным дисперсиям.

Сформулированную задачу решаем следующим образом:

1. Для различных служб и сервисов по результатам N наблюдений сетевого трафика сформируем выборку из N значений $x_1, x_2, ..., x_N$ исследуемой случайной величины X.
2. Для каждой выборки в соответствии с выражениями (1-2), рассчитываем значения выборочных средних (x^* и y^*) и дисперсий (σ_1^2 и σ_2^2).
3. Выбираем уровень значимости q, численно равный вероятности неприемлемых отклонений от принятой гипотезы и рассчитываем граничные точки F_1 и F_2 допустимых значений F.
4. Рассчитываем статистику теста F и проверяем условие $F_1 \le F \le F_2$.

5. При попадании значения F в критическую область гипотеза отвергается, а в случае непопадания – принимается.

Полученные результаты исследований позволят:

- экспериментально подтвердить или опровергнуть гипотезу об однородности свойств сетевого трафика различных служб и информационных сервисов методом дисперсионного анализа;

- обосновать практические рекомендации по организации программных и аппаратных средств мониторинга сетевой активности, обнаружения вредоносного программного обеспечения и предотвращения его воздействия на защищаемые инфокоммуникационные ресурсы.

4 Результаты экспериментальных исследований

Для проведения экспериментальных исследований свойств сетевого трафика были использованы эмпирические данные, полученные в результате работы программного анализатора (*снифера*) «Wireshark». Выбор этого программного сетевого анализатора связан с возможностью перехвата сетевого трафика в режиме реального времени. В ходе замеров с использованием «Wireshark» оценивался объем данных, передаваемых через компьютерную сеть за определённый период времени. Измерения объема трафика проводились как по числу пакетов, так и по числу бит данных. При этом эмпирические данные были получены и обобщены не менее чем по 100 000 временным отсчетам.

В качестве исходных данных при проведении экспериментальных исследований использовались различные телекоммуникационные службы и информационные сервисы:

- FTP (File Transfer Protocol) – стандартный протокол, предназначенный для передачи файлов по TCP-сетям;

- HTTP (HyperText Transfer Protocol) – протокол прикладного уровня передачи данных;

- электронная почта (E-mail) – технология обмена электронными сообщениями по распределённой (в том числе глобальной) компьютерной сети;

- Skype – программное решение, обеспечивающее текстовую, и аудиовизуальную связь посредством Интернет;

- YouTube – сервис, предоставляющий услуги видеохостинга.

Примеры гистограмм сетевого трафика при загрузке данных с сервиса YouTube (720p, бит/с), при использовании сервиса Skype в случае голосовой связи (voice) и видеосвязи (video), а также услуг электронной почты (E-mail) и протоколов HTTP и FTP приведены на рис. 2 – 7[1].

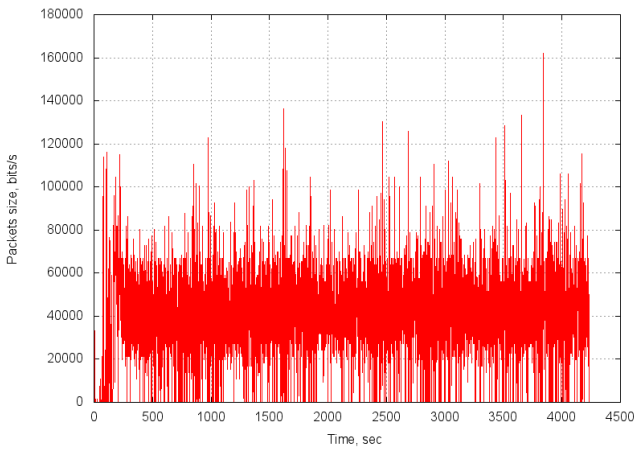


Рис. 2 - Фрагмент гистограммы сетевого трафика для YouTube

При проведении исследований использованы эмпирические данные по 100 временным отсчетам случайным образом выбранных отрезков сетевого трафика, соответствующих различным службам и сервисам. Т.е. оценка однородности сетевого трафика проводилась по выборочным данным с использованием основной метрики рассеивания – дисперсии случайной величины.

В соответствии с основными положениями центральной предельной теоремы теории вероятностей сумма достаточно большого количества слабо зависимых случайных величин, имеющих примерно одинаковые масштабы, имеет распределение, близкое к нормальному [14].

[1] Сетевой трафик представлен в виде числа бит данных переданных за 1 секунду.

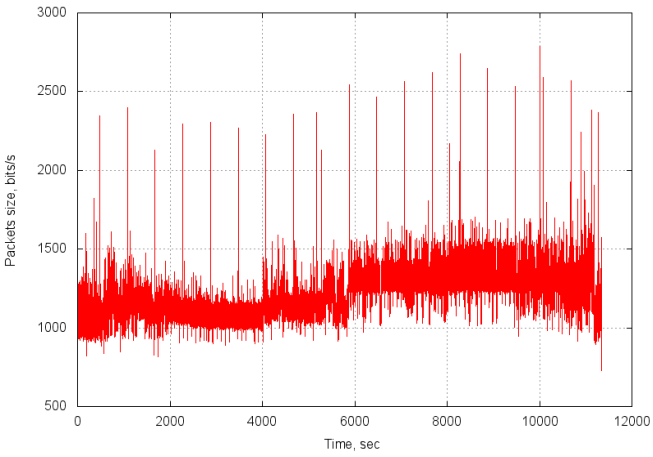


Рис. 3 - Фрагмент гистограммы сетевого трафика для Skype (voice, бит/с)

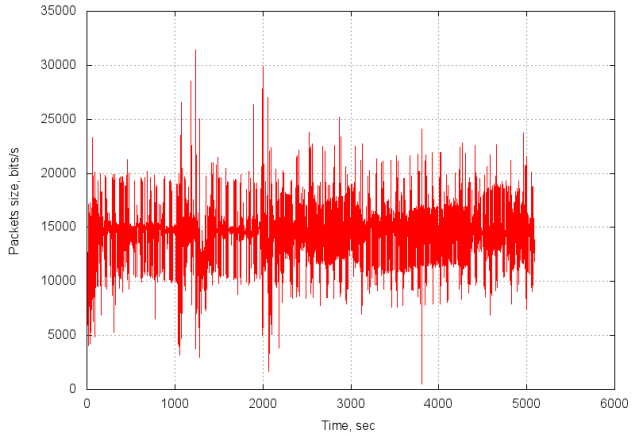


Рис. 4 - Фрагмент гистограммы сетевого трафика для Skype (video, бит/с)

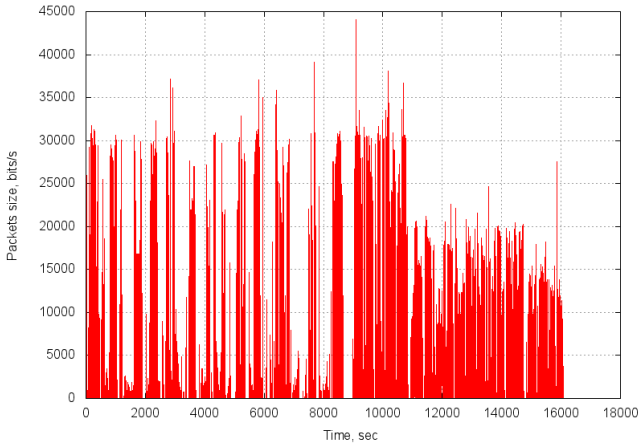


Рис. 5 - Фрагмент гистограммы сетевого трафика для E-mail (бит/с)

Так как объем данных, передаваемых через сетевую инфраструктуру за определённый период времени, является случайной величиной, формируемой под влиянием большого числа слабо зависимых случайных факторов, будем считать распределение этой случайной величины нормальным. При этом, естественно, должно соблюдаться условие, что ни один из факторов не является доминирующим при формировании сетевого трафика. Это предположение, в определенных случаях, может быть ошибочным, т.к. для некоторых служб и информационных сервисов телекоммуникационной сети существуют отдельные факторы, являющиеся доминирующими при формировании сетевого трафика (вносят основной вклад в объемы, данных передаваемых в единицу времени).

Принимая указанные предположения, воспользуемся аппаратом дисперсионного анализа для проверки статистической гипотезы об однотипности сетевых трафиков рассматриваемых служб и информационных сервисов моделируемой телекоммуникационной системы. Для этого выполним следующие основные этапы статистической проверки гипотез.

1. Сформулируем основную гипотезу H_0: - сетевые трафики однотипны по характеристике рассеивания, т.е. их выборочные дисперсии тождественны одной и той же генеральной дисперсии. Также сформулируем конкурирующую гипотезу H_1: - сетевые трафики не однотипны по характеристике рассеивания, т.е. их выборочные дисперсии не тождественны одной и той же генеральной дисперсии.

2. Зададим уровень значимости q, на в соответствии с которым в дальнейшем будет сделан вывод о справедливости гипотезы. Численно он равен вероятности допустить ошибку первого рода (*вероятности ложной тревоги*), т.е. вероятности отклонить гипотезу H_0, когда на самом деле она верна. Зададим уровень значимости равным q = 0,1.

3. Произведем расчет статистики теста так, чтобы: её величина зависела от исходной выборки; по её значению можно было бы сделать вывод об истинности гипотезы H_0; полу-

ченная статистика подчинялась бы известному и рассмотренному выше закону распределения Фишера.

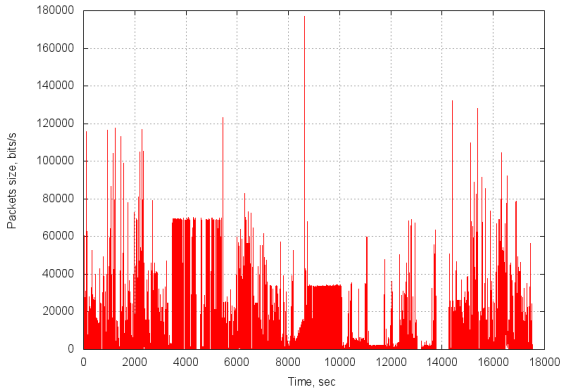


Рис. 6 - Фрагмент гистограммы сетевого трафика для HTTP (бит/с)

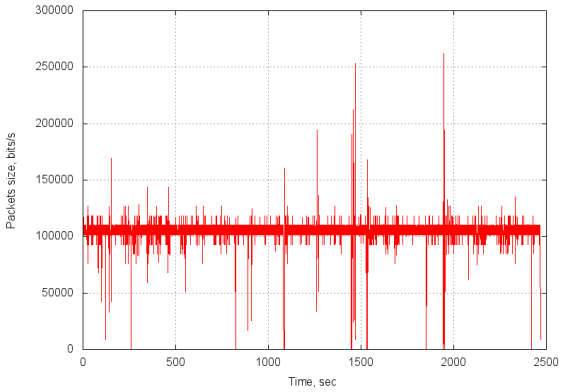


Рис. 7 - Фрагмент гистограммы сетевого трафика для FTP (бит/с)

4. Построим критическую область, т.е. зададим граничные точки F_1 и F_2 допустимых значений F, и из области значений статистики теста выделим подмножество значений (*критическую область*) $F < F_1$ и $F > F_2$, . По этим значениям будем судить о существенных расхождениях с предположением. Размер этой области определим из условия выполнения равенства $P(F < F_1 \vee F > F_2) = q = 0{,}1$.

5. Сделаем вывод об истинности гипотезы H_0. Для этого, по наблюдаемым значениям выборки, рассчитаем статистику теста и по попаданию (или непопаданию) в критическую область ($F < F_1 \vee F > F_2$) вынесем решение об отклонении (или принятии) выдвинутой гипотезы H_0.

Расчет статистики теста (этап 3) основывается на подсчете отношения выборочных дисперсий (*сумм квадратов, деленных на «степени свободы»*), эта статистика имеет распределение Фишера. Построим это распределение для заданных степеней свободы $k_1 = k_2 = N_2 - 1 = N_1 - 1 = 99$.

Зависимости плотности вероятностей $f_x(x)$ распределения Фишера и соответствующего интегрального распределения вероятностей $F_x(x)$ для значений $k_1 = k_2 = 99$ приведены на рис. 8-9 (*использован MathCad15*).

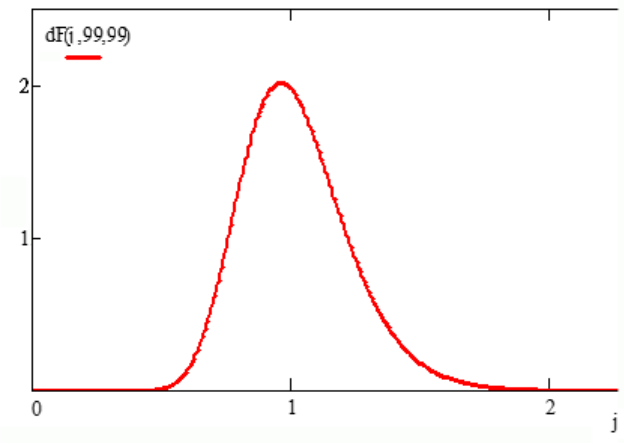


Рис. 8 - Зависимость плотности вероятностей распределения Фишера

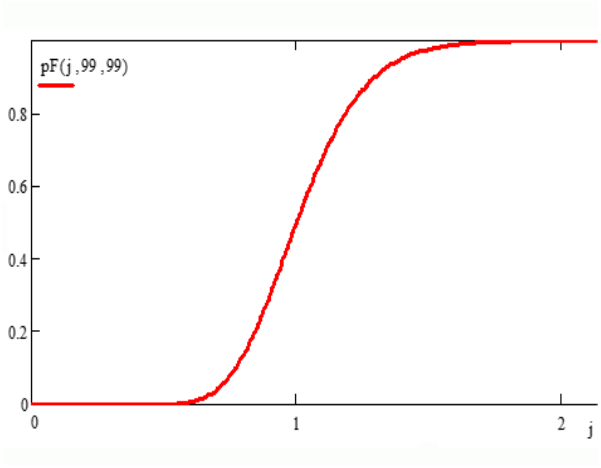


Рис. 9 - Зависимость интегрального распределения вероятностей Фишера

При решении практических задач часто требуется найти значение x, при котором функция распределения $F_x(x)$ случайной величины x принимает заданное значение p, т.е. требуется

решить уравнение $F_x(x) = p$. Решения такого уравнения (соответствующие значения x) в теории вероятностей принято называть *квантилями* [14,15].

Построим график обратного кумулятивного распределения вероятностей для заданного числа степеней свободы. Этот график описывает поведение квантили интегрального распределения вероятностей, т.е. поведение зависимости $x = F^{-1}{}_x(p)$. Для рассматриваемого случая, когда в качестве $F_x(x)$ используется интегральное распределение вероятностей Фишера с числом степеней свободы $k_1 = k_2 = 99$ (см. рис. 9), график обратного кумулятивного распределения имеет вид, приведенный на рис. 10.

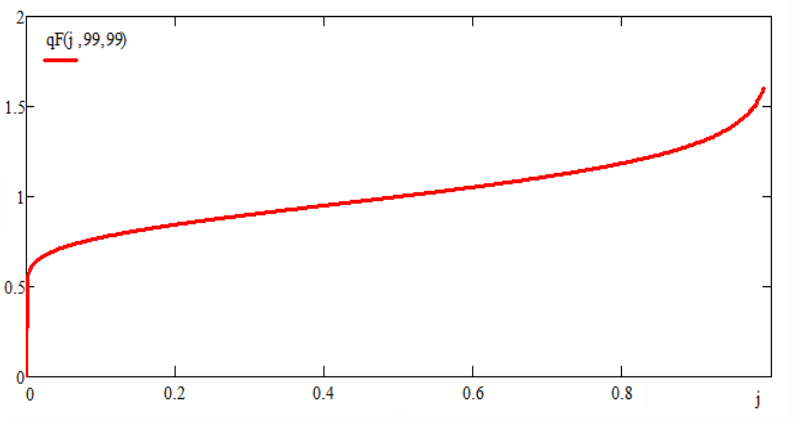


Рис. 10 - Зависимость обратного кумулятивного распределения вероятностей Фишера

Используя уровень значимости $q = 0.1$ с учетом зависимости на рис. 9, найдем такое значение правой граничной точки F_2 функции Фишера F, при котором $1 - F_x(F_2) < q/2 = 0.05$, что эквивалентно нахождению такой квантили $x = F_2$, при которой $x = F^{-1}{}_x(p = 1 - q/2)$, т.е. правая граничная точка F_2 определяется по правилу $F_2 = F^{-1}{}_x(0{,}95)$.

Найдем это значение и получим $F_2 = 1{,}394$, что наглядно подтверждается рис. 8–10. Таким образом, вероятность того, что значение F превысит правую граничную точку F_2 равна $q/2$:

$$P(F > F_2 = 1{,}394) = q/2 = 0{,}05.$$

Аналогично найдем значение левой граничной точки F_1, при котором $1 - F_x(F_1) < 1 - q/2 = 0.95$, что эквивалентно нахождению такой квантили $x = F_1$, при которой $x = F^{-1}{}_x(p = q/2)$, т.е. левую граничную точку F_1 определим по правилу $F_1 = F^{-1}{}_x(0{,}05)$.

Получим значение $F_1 = 0{,}717$, что наглядно демонстрируют зависимости на рис. 8–10. Очевидно, что вероятность того, что значение F не превысит левую граничную точку F_1 также равна $q/2$:

$$P(F < F_1 = 0{,}717) = q/2 = 0{,}05,$$

а вероятность попадания значения F в критическую область будет, соответственно, равна

$$P(F < F_1 = 0{,}717 \vee F > F_2 = 1{,}394) = 0{,}1.$$

Если значение рассчитанной на 3-м этапе статистики попадает в критическую область, т.е. лежит ниже левой или выше правой граничной точки, тогда гипотеза H_0 об однотипности исследуемых сетевых трафиков по характеристике их рассеивания отвергается, т.е. их выборочные дисперсии не тождественны одной и той же генеральной дисперсии.

Если же это значение не попадает в критическую область, т.е. лежит выше левой и ниже правой граничной точки, тогда гипотеза H_0 принимается, т.е. полагаем, что исследуемые сетевые трафики однотипны, их выборочные дисперсии тождественны одной и той же генеральной дисперсии.

Применим рассмотренный метод дисперсионного анализа к проверке гипотезы об однотипности различных трафиков, присущих различным службам и информационным сервисам телекоммуникационной сети. Для этого для каждого исследуемого трафика сформируем вы-

борку по 100 временным отсчетам данных, проведем обработку выборочных данных, т.е. проведем оценку выборочных средних и выборочных дисперсий по аналитическим выражениям (1-2). Полученные результаты экспериментальных исследований сведены в таблицу 1.

Таблица 1 - Результаты оценки данных для сетевых трафиков различных служб и сервисов

Вид трафика (служба, сервис)	Оценка выборочной дисперсии	Оценка выборочного среднего
YouTube (720p)	2,4×108	41372,8
Skype (voice)	14560,9	1154,9
Skype (video)	7,6×106	14738,7
E-mail	116079	122,5
HTTP	2,2×108	11567,8
FTP	2,4×108	104970

Полученные результаты дисперсионного анализа свидетельствуют о том, что статистический критерий на основе отношения выборочных дисперсий дает надежный механизм проверки однородности сетевого трафика. В частности, дисперсионный анализ с критерием значимости q = 0,1 позволяет верно определять используемые сервисы Skype и E-mail по выборочным наблюдениям из 100 временных отсчетов. Значения выборочных дисперсий для этих видов трафика существенно отличаются от значений выборочных дисперсий других сетевых служб и сервисов, в частности, от трафиков YouTube и протоколов HTTP, FTP.

В тоже время показатели рассеивания статистических данных для трафиков сервиса YouTube и протоколов HTTP и FTP очень близки. Статистика теста, полученная на основе расчета отношения выборочных дисперсий для них значительно отличается, что свидетельствует об однородности соответствующих данных. Практически это означает, что метод дисперсионного анализа не позволяет правильно различить эмпирические данные трафиков для YouTube, HTTP и FTP, они однородны по показателям статистического рассеивания.

5 Выводы

Проведенные исследования позволяют утверждать, что применение методов дисперсионного анализа позволяет подтвердить или опровергнуть гипотезу об однородности свойств сетевого трафика для различных телекоммуникационных служб и информационных сервисов. В частности, в ходе исследований по отношению выборочных дисперсий наблюдаемого сетевого трафика установлена разнородность соответствующих статистических данных. Этот факт позволяет с высокой вероятностью детектировать сетевую активность различных телекоммуникационных служб и информационных сервисов.

Избирательность сетевого детекта создает хорошие исходные условия для обнаружения несанкционированной сетевой активности и действий вредоносного программного кода, что способствует повышению общего уровня защиты инфокоммуникационных ресурсов телекоммуникационных систем и сетей.

Адаптация предложенных механизмов мониторинга сетевой активности с алгоритмами работы IDS и IPS систем является перспективным направлением для расширения возможностей систем сетевой защиты. В частности, полученные результаты могут быть полезны при построении новых механизмов обнаружения и предотвращения вторжений в перспективных Smart Grid системах.

Ссылки

[1] Cybersecurity for Smart Grid Systems URL: https://www.nist.gov/programs-projects/cybersecurity-smart-grid-systems (Last accessed: 25 June 2018)

[2] Dagle J. E. Cyber-physical system security of smart grids. 2012 IEEE PES Innovative Smart Grid Technologies (ISGT). Washington, DC, 2012. P. 1 – 2.
[3] Lightweight Stream Ciphers for Green IT Engineering / Kuznetsov O. and all. Green IT Engineering: Social, Business and Industrial Applications. Studies in Systems, Decision and Control. 2019. Vol. 171. P. 113 – 137.
[4] Christodorescu M., Jha S. Testing. Malware Detectors. Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA'04). Boston: Massachusetts, USA, 2004. 11 p.
[5] Prospective Lightweight Block Cipher for Green IT Engineering /Andrushkevych A. and all. Green IT Engineering: Social, Business and Industrial Applications. Studies in Systems, Decision and Control. 2019. Vol. 171. P. 95 – 112.
[6] Methods of Information Protection in Communications Systems and Methods of Their Cryptoanalysis/ Gorbenko I.D., Dolgov V.I., Rublinetskii V.I., Korovkin K.V. Telecommunications and Radio Engineering. 1998. Vol. 52, Issue 4. P. 89 – 96.
[7] OpenStack-Based Evaluation Framework for Smart Grid Cyber Security / Albarakati A. and all. 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm). Aalborg, 2018. P. 1 – 6.
[8] Jahan S., Habiba R. An analysis of smart grid communication infrastructure & cyber security in smart grid. 2015 International Conference on Advances in Electrical Engineering (ICAEE). Dhaka, 2015. P. 190 – 193.
[9] Smart grid information security - a research on standards/ Wang Y., Zhang B., Lin W., Zhang T. 2011 International Conference on Advanced Power System Automation and Protection. Beijing, 2011. P. 1188 – 1194.
[10] Security and Reliability Perspectives in Cyber-Physical Smart Grids / Lei H., Chen B., Butler-Purry K. L., Singh C. 2018 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia). Singapore, 2018. P. 42 – 47.
[11] Smart grid information security - a research on standards / Wang Y., Zhang B., Lin W., Zhang T. 2011 International Conference on Advanced Power System Automation and Protection. Beijing, 2011. P. 1188 – 1194.
[12] Impact of cyber-security issues on Smart Grid / Yang Y and all. 2011 2nd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies. Manchester, 2011. P. 1 – 7.
[13] Zamula A., Kavun S. Complex systems modeling with intelligent control elements. International Journal of Modeling, Simulation, and Scientific Computing. 2017. Vol. 8, № 1. [19 pages]
[14] Zetter K. The Ukrainian Power Grid Was Hacked Again. Vice Motherboard. URL: https://motherboard.vice.com (Last accessed: January 10, 2017)
[15] Lipovsky R. New wave of cyberattacks against Ukrainian power industry. We Live Security. URL: https://www.welivesecurity.com (Last accessed: 20 February 2016)

Reviewer: Volodymyr Maxymovych, Doctor of Sciences (Eng.), Full Prof., Institute of Computer Technologies, Automation and Metrology, Lviv Polytechnic National University, 79013, Ukraine.
E-mail: vmax@polynet.lviv.ua

Received on Aprilr 2019.

Authors:

Alexandr Kuznetsov, Doctor of Sciences (Eng.), Full Prof., Academician of the Academy of Applied Radioelectronics Sciences, V.N.Karazin Kharkiv National University, Kharkov, Ukraine.
E-mail: kuznetsov@karazin.ua

Vlada Hryhorenko, computer science student, V.N. Karazin Kharkiv National University, Ukraine.
E-mail: wdpgames@yandex.ru

Andrii Diachenko, computer science student, V.N. Karazin Kharkiv National University, Ukraine.
E-mail: andrey.090220@gmail.com

Mykhaylo Bagmut, postgraduate, V.N. Karazin Kharkiv National University, Ukraine.
E-mail: sapsanmiha@gmail.com

Dispersion analysis of network traffic for intrusion detection in Smart Grids.

Abstract. We consider the systems of detection and prevention of intrusions in modern telecommunication systems and networks. Methods of monitoring events, consisting of analysis of network activity of individual services and information services, are analyzed. It is proposed to use the mathematical apparatus of analysis of variance for processing results of modeling telecommunication systems and studying the statistical properties of network traffic in determining the significance of discrepancies or coincidence of characteristics. The proposed approach is to use the Fisher statistical criterion based on an estimate of the ratio of sample variances. This allows you to test the hypothesis about the homogeneity of statistical properties of network traffic with respect to the variance index (variance) with a given level of significance. The obtained results of experimental studies are recommended to be used to improve mechanisms for monitoring the network activity of individual services and information services, including for detecting and preventing intrusions in telecommunications systems and networks of promising Smart Grids.

Keywords: Telecommunication Systems and Networks; Intrusion Detection and Prevention System; Analysis of Variance; Cyber Security; Smart Grid.

Рецензент: Володимир Максимович, д.т.н., проф., Інститут комп'ютерних технологій, автоматики та метрології Національного університету «Львівська політехніка», Львів, 79013, Україна.
E-mail: vmax@polynet.lviv.ua

Поступила: Апрель 2019.

Автори:

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

Влада Григоренко, студентка факультету комп'ютерних наук, Харківський національний університет ім. В.Н. Каразіна, Харків, Україна.
E-mail: mrsgriggy@gmail.com

Андрій Д'яченко, студент факультету комп'ютерних наук**,** Харківський національний університет ім. В.Н. Каразіна, Харків, Україна
E-mail: andrey.090220@gmail.com

Михайло Багмут, аспірант, Харківський національний університет ім. В.Н. Каразіна, Харків, Україна.
E-mail: sapsanmiha@gmail.com

Дисперсійний аналіз мережевого трафіку для виявлення вторгнень в Smart Grids.

Анотація. Розглядаються системи для виявлення та запобігання вторгнень у сучасних телекомунікаційних системах і мережах. Досліджуються методи моніторингу подій, що базуються на аналізі мережевої активності окремих служб та інформаційних сервісів. Пропонується використовувати математичний апарат дисперсійного аналізу для обробки результатів моделювання телекомунікаційних систем і дослідження статистичних властивостей мережевого трафіку при визначенні значущості розбіжності або збігу характеристик. Пропонований підхід полягає у використанні статистичного критерію Фішера, заснованого на оцінці відношень вибіркових дисперсій. Це дозволяє з заданим рівнем значущості перевіряти гіпотезу про однорідність статистичних властивостей мережевого трафіку щодо показника розсіювання (дисперсії). Отримані результати експериментальних досліджень рекомендується використовувати для вдосконалення механізмів моніторингу мережевої активності окремих служб та інформаційних сервісів, в тому числі для виявлення і запобігання вторгнень у телекомунікаційних системах та мережах перспективних Smart Grids.

Ключові слова: телекомунікаційні системи та мережі; системи для виявлення та запобігіння вторгнень; дисперсійний аналіз; кібербезпека; Smart Grids.

УДК 004.415.53 DOI: 10.26565/2519-2310-2019-1-07

АНАЛИЗ ИНСТРУМЕНТОВ ДЛЯ АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Ольга Мелкозерова, Алексей Нарежний, Сергей Малахов

Харьковский национальный университет имени В.Н. Каразина, Харьков, 61022, Украина
olja.mex@gmail.com, o.nariezhnii@karazin.ua, mailgate@meta.ua

Рецензент: Вячеслав Калашников, д.ф.-м.н., проф., Технологический университет Монтеррея, 64849 Монтеррей, Нуево-Леон, Мексика
kalash@itesm.mx

Поступила: Май 2019.

Аннотация: *Тестирование качества программного обеспечения (ПО) является трудоемким и ответственным этапом его разработки. Это обуславливает практический интерес к автоматизации основных этапов тестирования. Как показывает практика, использование всевозможных инструментов тестирования ПО заметно облегчает этот процесс. Однако, принципиально важно применять тот или иной инструмент, учитывая специфику каждого конкретного случая. Это обстоятельство обусловлено большими объемами тестируемой информации и сложностью эксплуатационной документации. В работе приведен обзор и анализ возможностей существующих инструментов для проведения автоматизированного тестирования ПО, с описанием соответствующих технологий, назначения и области применения. Приведены примеры составления тестов с использованием Selenium, SerenityBDD и JMeter.*

Ключевые слова: *автоматизированное тестирование; инструменты автоматизированного тестирования; технологии автоматизированного тестирования.*

1 Введение

Тестирование программного обеспечения (ПО) является важнейшим этапом в ходе его разработки. Как известно, процесс тестирования ПО можно разделить на мануальное (выполняется вручную) и автоматизированное. Для автоматизированного тестирования ПО используются различные подходы и инструменты, которые способны частично заменить участие человека при проведении значительного количества рутинных процедур. В настоящее время существуют и в той или иной мере используется большое количество подобных программных инструментов. В связи с этим, в данной роботе представлен обзор соответствующих решений (инструментов и технологий) с кратким описанием их назначения и целевой областью применения. Кроме того, профильные специалисты выделяют еще и виды тестирования, которые следует рассматривать, как список действий, которые необходимо совершить при тестировании того или иного ПО.

2 Технологии автоматизированного тестирования

К технологиям автоматизированного тестирования относят [1]:

1) частные решения;
2) тестирование под управлением данными (*DDT - data driven testing*);
3) тестирование под управлением ключевыми словами (*KDT - Keyword Driven Testing*);
4) использование фрéймворков (*Frameworks*);
5) запись и воспроизведение (*Record & Playback*);
6) поведенческое тестирование (*BBT - Behaviour Driven Testing*).

Следует отметить, что один и тот же инструмент тестирования может использовать сразу несколько технологий автоматизированного тестирования. Например, фреймворк Selenium IDE [2], в настоящее время по своему прямому назначению практически не используется, однако очень удобен для обучения соответствующих специалистов. В данном случае, необходимо буквально несколько минут, для того чтобы человек, незнакомый с автоматизированным тестированием, написал свой первый тест. Selenium IDE (*Integrated Development Environment*) использует технологию записи и воспроизведения (*Record & Playback*), однако все

его команды (например, «open» или «verifyText») – это уже тестирование под управлением ключевыми словами (*KDT)*.

В общем случае в связи с появлением фрейворков, ускорился процесс обучения специалистов в области обеспечения качества создаваемого ПО. Использование этой технологии сводится к следующим основным шагам:

1. Тестировщик ПО вручную выполняет тест-кейс, а средство автоматизации его записывает. Тест-кейс можно сохранить в html-формате (рис. 1).
2. Результаты записи представляются в виде кода автоматизированного теста на высокоуровневом языке программирования (*в некоторых случаях специально разработанном*).
3. Тестировщик редактирует полученный код.
4. Готовый код выполняется для проведения тестирования в автоматизированном режиме.

1_01		
open	https://www.olx.ua/uk/changelang/?lang=ru&l=https%3A%2F%2Fwww.olx.ua	
verifyText	css=span.link.inlblk > strong	Мой профиль
verifyText	//div[@id='lastwrapper']/div/div[2]/div/div[2]/ul/li[5]/a/span	Популярные запросы

Рис. 1 – Тест-кейс *(вариант)*

К инструментам Selenium следует отнести также: Selenium WebDriver и Selenium Server.

Selenium WebDriver – набор библиотек для различных языков программирования, позволяющий управлять браузером из программы, написанной на этом языке программирования. Представляет собой надежный фреймворк автоматизации, способный работать с любым браузером. Кроме того, позволяет разработать большой тестовый набор, включающий тесты с достаточно сложной логикой поведения и проверок.

Selenium Server – может принимать команды от удаленного компьютера, где работает сценарий автоматизации и исполнять их в браузере. Несколько серверов Selenium могут формировать распределенную сеть, которая называется Selenium Grid, что обеспечивает масштабирование стенда автоматизации.

Selenium поддерживает команды трех видов [2]: - действия (*actions*); - считыватели (*accessors*); - проверки (*assertions*).

Действия – это команды, которые, как правило, управляют состоянием приложения. Они совершают процедуры вроде «щелкнуть по той ссылке» или «выбрать эту операцию». Если действие не может быть выполнено, либо выполняется с ошибкой, то текущий тест прерывается.

К большей части действий можно добавить суффикс «AndWait» («подождать»), «ClickAndWait». Этот суффикс сообщает Selenium, что действие принудит браузер совершить запрос к серверу и что Selenium должен дождаться загрузки новой страницы.

Считыватели анализируют состояние приложения и сохраняют результаты в переменные *(к примеру, команда «storeTitle»)*. Кроме того, они используются для автоматической генерации проверок.

Проверки похожи на «Считыватели», однако, в отличие от них, проверяют соответствие текущего состояния приложения, ожидаемому. Например, удостовериться, что заголовок страницы имеет определенное название.

Технологию записи и воспроизведения (*Record & Playback*) использует также инструмент для тестирования производительности JMeter [3], инструмент UI Katalon Studio [4] и другие фреймворки.

3. Уровни тестирования

К уровням тестирования относят [5,6]:

- компонентное тестирование (модульное), сфокусированное на тестировании индивидуальных компонентов (*unit testing*);

- интеграционное тестирование. Выявляет дефекты интерфейса и ошибки между взаимодействующими компонентами или системами (*integration testing*);
- системное тестирование (*system testing*). Проверяет взаимосвязанные системы для верификации указанных требований (*выполняется инженерами по контролю качества*);
- приемочное тестирование (*acceptance testing*). Верифицирует соблюдение с требованиями, бизнес процессами и пользовательскими пожеланиями.

Модульное тестирование нацелено на поиск возможных дефектов и верификацию функционирования программных модулей, программ, объектов, классов [1], которые можно протестировать изолированно.

Компонентное тестирование может включать в себя, как тестирование функциональности и специфичных нефункциональных характеристик (таких как поведение ресурсов (*например, поиск «утечки» памяти*)) или тестирование надежности, так и структурное тестирование (*например, покрытие кода*). На практике компонентное тестирование, обычно, выполняется разработчиками, которые непосредственно пишут код. При этом, выявленные дефекты обычно исправляются сразу же, без занесения их в базу дефектов.

Для тестирования на этом уровне можно использовать семейство библиотек *XUnit*, к примеру для языка программирования Java следует использовать *TestNG* и *JUnit* - фреймворки для написания повторяющихся модульных тестов. Можно подключить через зависимости [7]. Зависимости – это те библиотеки, которые непосредственно используются в данном проекте для целей компиляции кода или его тестирования.

JUnit обеспечивает перегруженные методы для всех примитивных типов, объектов и массивов. Параметры: - ожидаемое значение и актуальное значение. Опционально первым параметром может быть сообщение, которое появляется, если тест «провален». Например, приведенная ниже строка проверяет, что два примитива равны:

Assert.assertEquals(expectedResult,result,0) .

Уровни интеграционного тестирования:

- компонентное интеграционное тестирование проверяет взаимодействие между программными компонентами и производится после компонентного тестирования;
- системное интеграционное тестирование проверяет взаимодействие между системами или между аппаратным обеспечением и может быть выполнено после системного тестирования.

4. Тестирование под управлением поведением (BBT)

Все технологии автоматизированного тестирования ПО имеют одну общую особенность: - они сфокусированы на технических аспектах поведения приложений. Кроме того, они обладают и общим недостатком: - с их помощью сложно проверить высокоуровневые пользовательские сценарии. Этот недостаток призвано исправить тестирование под управлением поведением, при котором акцент делается не на отдельных технических деталях, а на общей работоспособности приложения при решении типичных пользовательских задач.

Такой подход не только упрощает выполнение целого ряда проверок, но и облегчает взаимодействие между разработчиками, тестировщиками, бизнес-аналитиками и заказчиками ПО. В его основе лежит формула «*given – when – then*» [1].

Given (“имея, предполагая, при условии”) описывает начальную ситуацию, в которой находится пользователь в контексте его работы с приложением.

When (“когда”) описывает набор действий пользователя в данной ситуации.

Then (“тогда”) описывает ожидаемое поведение приложения.

Примером использования этой технологии является *Serenity BDD* [8].

Код для теста выглядит следующим образом (см. рис. 2):

```
@Test
public void verifySubCategory() throws InterruptedException {
    // GIVEN
    stepsForSerenity.a_user_visits_a_page(siteHomePage);
    // WHEN
    stepsForSerenity.the_user_chooses_category_Compurer();
    // THEN
    stepsForSerenity.the_user_can_see_subcategory_ITService("IT услуги");
}
```

Рис. 2 – Код теста для Serenity BDD *(пример)*

Один из шагов Serenity BDD:

```
@Step("Given the user visits a page {0}")
public void a_user_visits_a_page(String homePage) {
    this.siteHomePage = homePage;
}
```

Рис. 3 – Пример шага для Serenity BDD.

Также можно сгенерировать отчет (см. Рис. 4):

Steps	Outcome	Duration
Given the user visits a page https://pn.com.ua/	SUCCESS	0,05s
When the user chooses category Compurer	SUCCESS	37,49s
Then the user sees IT услуги subcategory	SUCCESS	1,62s
	SUCCESS	*39,51s*

Рис. 4 – Отчет о прохождении теста для Serenity BDD.

5. Тестирование производительности

Тестирование производительности (*performance testing*) – исследование «скоростных показателей» приложения при различной (по характеру и количественных показателях) нагрузке [1].

Нагрузочное тестирование (*load testing*) – исследование способности приложения сохранять заданные показатели качества при нагрузке в допустимых пределах и некотором превышении этих пределов («запас прочности»).

Стрессовое тестирование (*stress testing*) – исследование поведения приложения при не декларированных изменениях нагрузки.

Объемное тестирование (*volume testing*) – исследование производительности системы при обработке различных объемов данных.

Тестирование масштабируемости (*scalability testing*) – исследование способности приложения увеличивать показатели производительности в соответствии с увеличением количества доступных данному приложению ресурсов.

Конкурентное тестирование (*concurrency testing*) – исследование поведения приложения, в случаях, когда ему приходится обрабатывать большое количество одновременно поступающих запросов, что вызывает конкуренцию между этими запросами за ресурсы (*базу данных, память, канал передачи данных, дисковую систему и др.*). Иногда под конкурентным тестированием понимают, также исследование работы многопоточного приложения и корректность синхронизации действий.

Тестирование надежности (*reliability testing*) - тестирование способности приложения выполнять свои функции в заданных условиях на протяжении заданного времени или заданного количества операций.

Тестирование восстанавливаемости (*recoverability testing*) – тестирование способности приложения восстанавливать свои функции и заданный уровень производительности, а также восстанавливать свои данные в случае возникновения критической ситуации, приводящей к временной или частичной утрате работоспособности приложения.

Тестирование отказоустойчивости (*failover testing*) – эмуляция или реальное создание критической ситуации, с целью проверки способности приложения задействовать соответствующие встроенные механизмы, обеспечивающие предотвращение нарушения работоспособности, производительности и повреждения данных.

Элементы тест плана *JMeter* [9]:

- группы потоков (*Thread groups*);
- логические контролеры (*Logic controller*);
- типовые контроллеры (*Samle generating controller*);
- слушатели (*Listeners*);
- таймеры (*Timers*);
- соответствия (*Assertions*);
- конфигурационные элементы (*Configuration Elemetents*).

Группы потоков – начальные точки любого тест-плана (рис. 5). Все контроллеры и образцы должны быть в группе потоков. Другие элементы, такие как слушатели, могут располагаться под тест-планом, в котором они применяются для всех потоков групп. Элемент группы потоков управляет количеством потоков, который *JMeter* будет использовать для выполнения теста. В общем случае можно установить:

- количество потоков (пользователей) (*Number of Threads*);
- время наращивания (*Rump-Up Period*);
- количество повторов для выполнения теста.

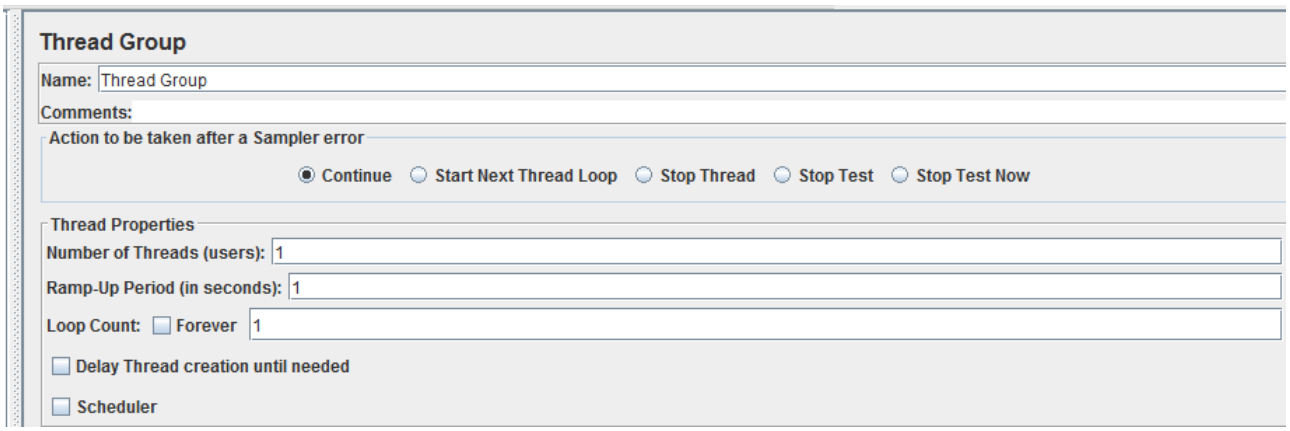


Рис. 5 – Группы потоков *Thread groups*.

Каждый поток будет исполнять свой тест-план полностью и абсолютно независимо от других тестовых потоков. Произведение потоков используется для моделирования конкурентных соединений для сервера приложения.

Rump-Up Period «сообщает» *JMeter*, сколько времени потребуется, чтобы нарастить количество потоков до полного количества. Например: - если используется 10 потоков, а Rump-*Up Period* 100 секунд, то *JMeter* «возьмет» 100 секунд для того, чтобы получить все 10 потоков. При этом каждый поток будет начинаться через 10 секунд (100/10) после того, как запустился предыдущий поток. Если есть 30 потоков, и время наращивания составляет 120 секунд, то каждый последующий поток будет задерживаться на 4 секунды.

Loop count – количество циклов теста, *forever* – тест будет длиться вечно.

Время наращивания должно быть достаточно длинным, чтобы избежать слишком больших нагрузок в начале теста и достаточно коротким, чтобы последние потоки начинали запускаться до того, как закончились первые.

Thread group также предоставляет планировщик *Scheduler* (рис. 6). Следует кликнуть на соответствующую кнопку, чтобы получить доступ к дополнительным полям, в котором пользователь может установить: - продолжительность (*Duration*) теста; - задержку запуска (*Startup Delay*); - время начала и окончания теста. Также пользователь может выбрать продолжительность и задержку, чтобы контролировать каждый поток. Здесь можно запускать тест в определенное время, например, ночью.

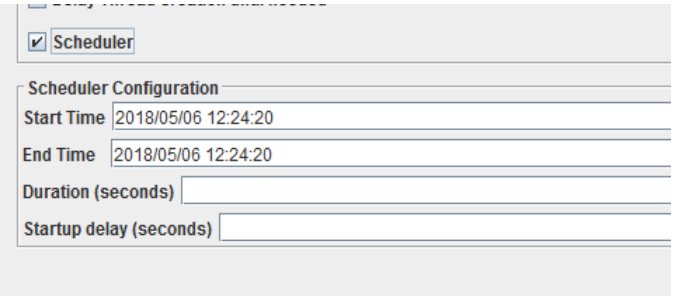


Рис. 6 – Группы потоков *Thread groups* *Scheduler* (*запуск теста ночью*)

JMeter имеет два типа контроллеров: типовые контроллеры (*Samplers*) и логические контроллеры (*Logical Controllers*). Они управляют процессом теста.

Типовые контроллеры отправляют запрос на сервер. Например: - добавить HTTP запрос, если вы хотите отправить HTTP запрос. Пользователь также может настроить запрос добавлением конфигурационных элементов к типовым контроллерам.

Логические контроллеры позволяют настроить логику, которую *JMeter* использует, когда отправляет запрос. Например: - пользователь может добавить *Interleave Logic Controller* в качестве альтернативы между двумя *HTTP* запросами типовых контроллеров.

Listener – это компонент, который показывает результаты образцов. Результаты могут быть представлены в дереве, таблице, графах или быть записаны в лог-файл.

Graph result listener – формирует простой граф (см. рис. 7-13). Зависимости, представленные на рис. 13-14 отображают следующие параметры (*см. цветовую маркировку*):

- данные - Data (черный цвет);
- среднее значение времени - Average (голубой цвет);
- стандартное отклонение - Deviation (красный цвет);
- пропускную способность - Throughtput (зеленый цвет) и медиану (Mediane).

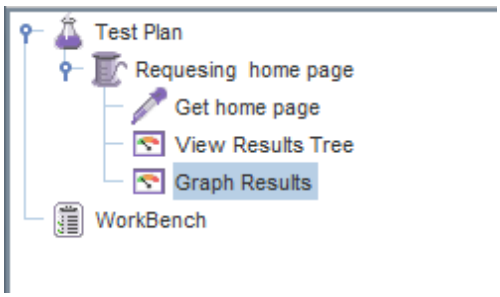


Рис. 7 – Фрагмент тест-плана для *Apache JMeter*

Пропускная способность отображает реальное количество запросов в минуту, которое обрабатывает сервер.

Составим наш тест-план (рис. 5). В него добавим *Thread Group - Requesting home page* (рис. 8) и *HTTP* запрос (рис. 9). Затем добавляем «Слушатели» (*Listeners*) *View Result* и *Tree Graph Results*. Сохраняем тест-план и запускаем на исполнение.

Tree Graph Results – дерево ответов всех образцов, позволяют пользователю увидеть отклик для всех образцов. В дополнение к показу отклика, можно видеть время этого отклика (*Sampler Result* – рис. 10) и некоторые коды для этих откликов (например, в *HTML* формате – рис. 11).

После запуска тест-плана, можно увидеть ответ на запрос (рис. 10 и 11), сообщение (отклик 200 ок). При этом если с сайтом возникнут проблемы, то сообщение будет выглядеть следующим образом – рис. 12. Результат запуска тест-плана при выполнении *Graph Result* представлен на рис. 13.

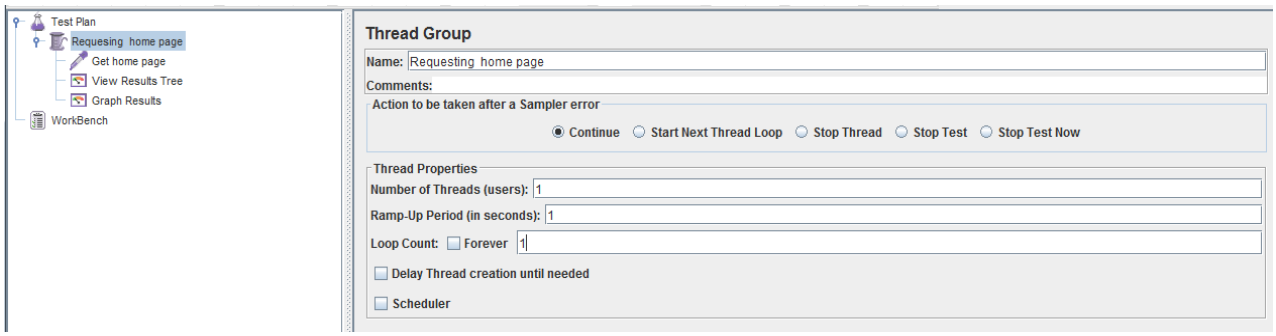


Рис. 8 – Изменение количества пользователей *Apache JMeter*

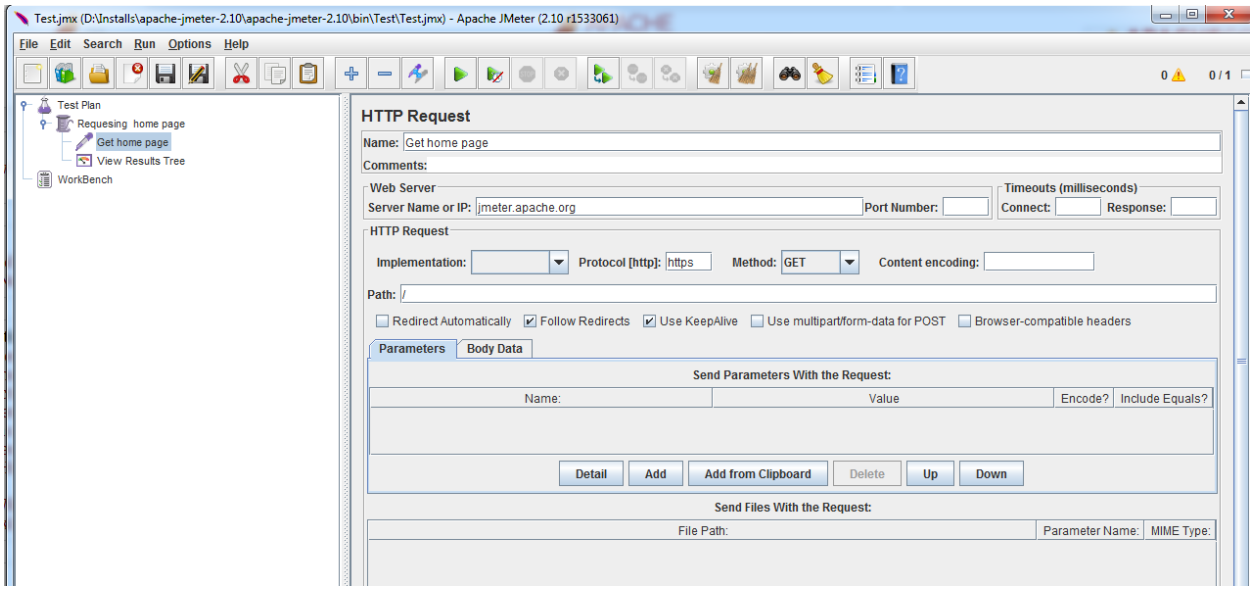


Рис. 9 – Заполнение *HTTP* запроса

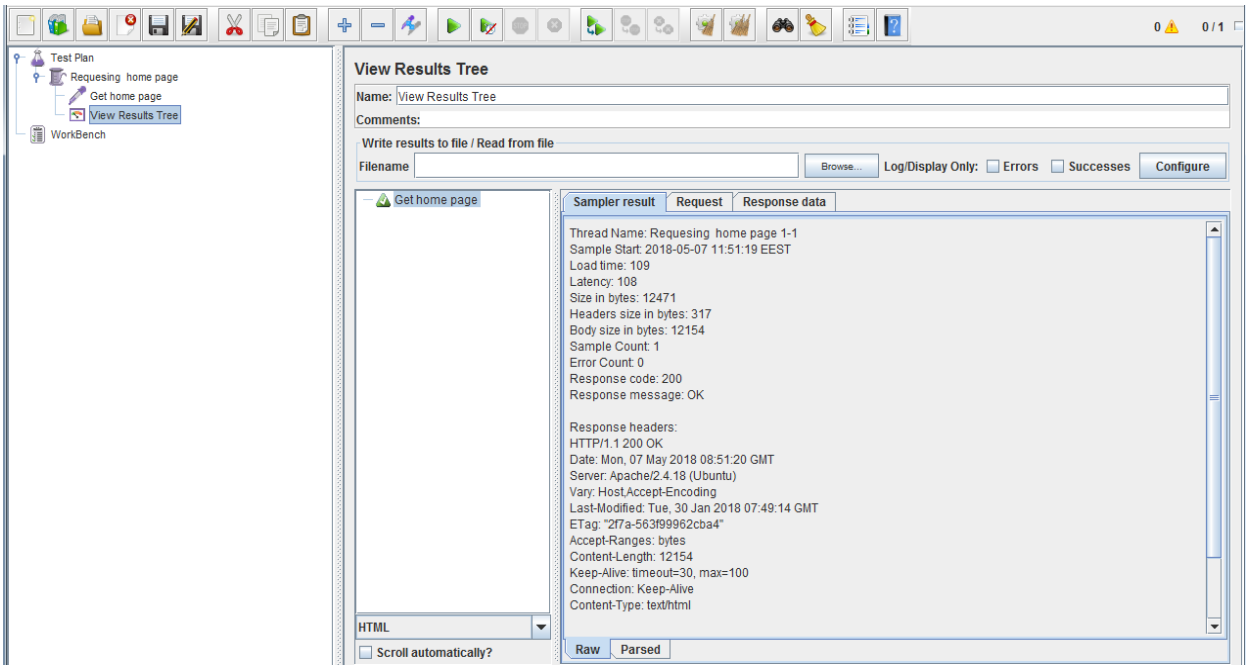


Рис. 10 – Результат запуска тест-плана при выполнении *View Result Tree*

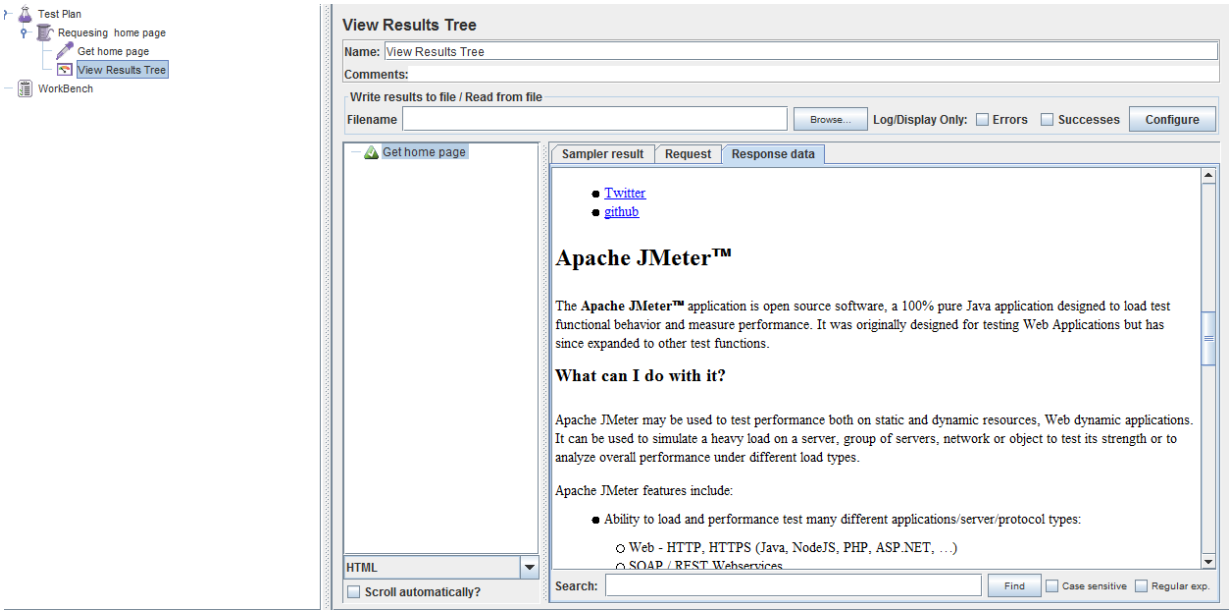


Рис. 11 - Результат запуска тест-плана при выполнении *View Result Tree*

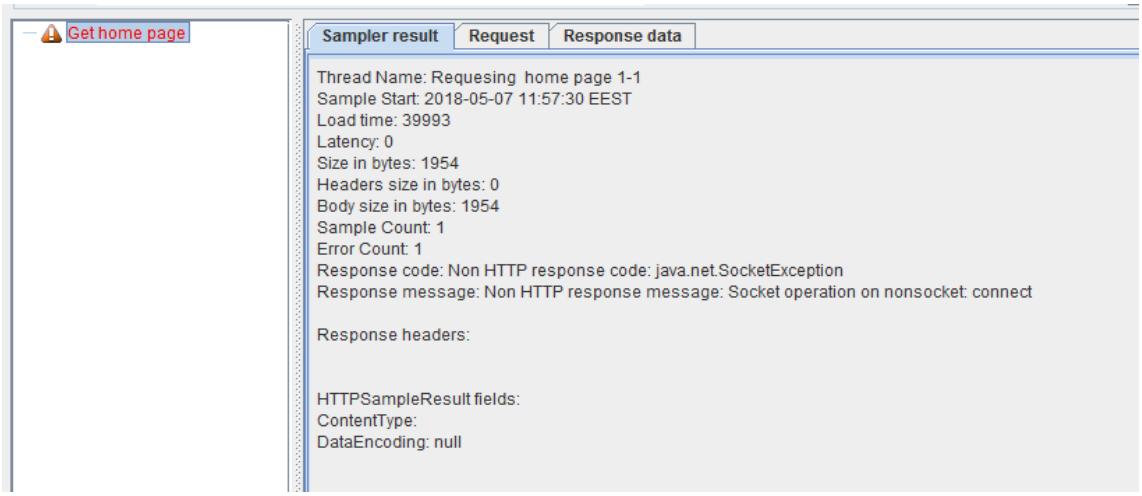


Рис. 12 - Результат запуска тест-плана при выполнении *View Result Tree*

Рис. 13 – Результат запуска тест-плана при выполнении *Graph Result*

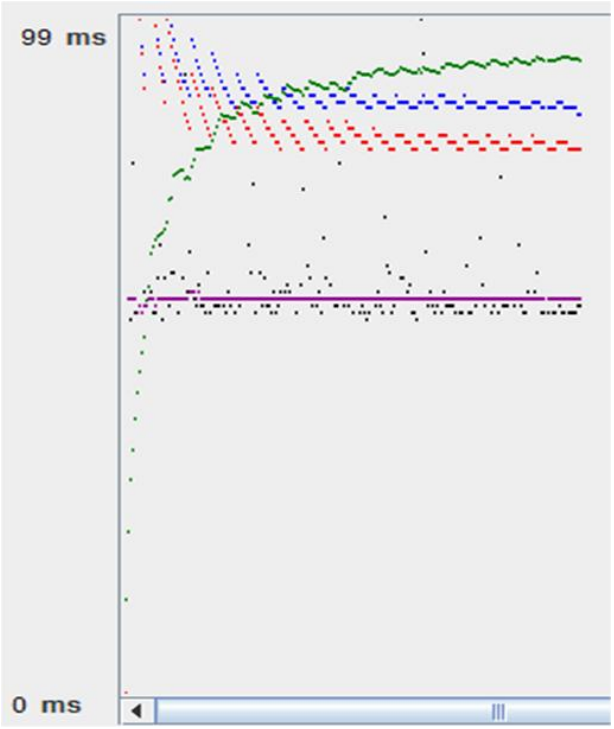


Рис. 14 – Фрагмент интерфейса рабочего окна «*Graph Results*» с результирующими зависимостями

5 Выводы

Автоматизированное тестирование, по сравнению с ручным, обладает целым рядом ощутимых преимуществ, где основные - это быстрота выполнения и отсутствие ошибок, обусловленных исключением человеческого фактора.

Тестирование производительности ПО - такой вид тестирования, где применима только автоматизация. Она также неизбежна на этапе разработки программного обеспечения. Но при ее применении также есть и недостатки, это большое количество времени на написание автотестов, их поддержку, необходим более квалифицированный персонал.

На этапе разработки ПО для модульного (компонентного) тестирования используют различные библиотеки тестирования (*JUnit, TestNG* и другие).

Для тестирования всей системы используют фреймворки, их существует большое множество. Например, для тестирования производительности сайтов используется *Jmeter*, который, в свою очередь, также позволяет тестировать *API* (*Application programming interface*). У данного вида тестирования есть определенные преимущества, в сравнении с *UI* тестированием:

1. Раннее тестирование – разработчики сначала делают *API*, а потом уже графический интерфейс. При этом существует возможность проверить логику раньше, чем дорисуют *GUI*.
2. Графического интерфейса может в принципе и не быть.
3. Скорость – вызов одного запроса занимает долю секунды. А вот через интерфейс повторить процедуру часто бывает сложно.
4. Автоматизация – даже если нет автотестов на уровне *API* приложения, всегда можно их создать вручную через *PostMan* или *Jmeter*.

На пользовательском уровне при проведении приемочного тестирования, хорошо себя зарекомендовали решения под управлением поведением, например: - *Serenity BDD* или *Cucumber*.

Ссылки

[1] Куликов С. Тестирование программного обеспечения. Базовый курс. URL: http://svyatoslav.biz/software_testing_book/ (дата звернення: 28.12.2017)
[2] SeleniumIDE. URL: https://www.seleniumhq.org/docs/ (дата звернення: 16.06.2019)
[3] Jmeter. URL: https://jmeter.apache.org/ (дата звернення: 16.06.2019)
[4] Katalon. URL: https://www.katalon.com/ (дата звернення: 16.06.2019)
[5] Основи инженеренерии качества / Андон Ф.И. та ін. Киев: Академпериодика, 2007. 672 с.
[6] Стандартный глоссарий терминов, используемых в тестировании программного обеспечения. International Software Testing Qualifications Board. – 2014. URL: https://docplayer.ru/340620-Standartnyy-glossariy-terminov-ispolzuemyh-v-testirovanii-programmnogo-obespecheniya.html
[7] Maven Reposotory. URL: https://mvnrepository.com / (дата звернення: 16.06.2019)
[8] Serenity. URL: http://www.thucydides.info/docs/serenity / (дата звернення: 16.06.2019)

Рецензент: В'ячеслав Калашников, д.ф.-м.н., проф., департамент систем і промислового виробництва Технологічного університету Монтеррея, Монтеррей, Мексика. E-mail: kalash@itesm.mx

Надійшло: Травень 2019.

Автори:
Ольга Мелкозьорова, к.т.н., старший викладач кафедри Безпеки інформаційних систем і технологій, Харківський національний університет імені В.Н. Каразіна, Харків, Україна**.**
E-mail: olja.mex@gmail.com
Олексій Нарєжній, к.т.н., доцент кафедри Безпеки інформаційних систем і технологій, Харківський національний університет імені В.Н. Каразіна, Харків, Україна.
E-mail: o.nariezhnii@karazin.ua
Сергій Малахов, к.т.н., с.н.с., Харківський національний університет імені В.Н. Каразіна, Харків, Україна.
E-mail: mailgate@meta.ua

Аналіз інструментів для автоматизованого тестування програмного забезпечення.
Анотація. Тестування якості програмного забезпечення (ПЗ) є дуже трудомістким та відповідальним етапом при його розробці. Це обумовлює практичний інтерес до автоматизації основних процедур при тестуванні. Як показує практика, використання можливих інструментів тестування ПЗ суттєво полегшує цей процес. Але принципово важливо застосовувати той чи інший інструмент, враховуючи специфіку кожного окремного випадку. Ця обставина обумовлена великою кількістю інформації, що тестується, та складністю експлуатаційної документації. У роботі наведено огляд та аналіз можливостей існуючих інструментів, для проведення автоматизованого тестування ПЗ, з описом відповідних технологій, призначенням та області використання. Наведено приклади складання тестів з використанням Selenium, Serenity BDD и JMeter.

Ключові слова: автоматизоване тестування; інструменти автоматизованого тестування; технології автоматизованого тестування.

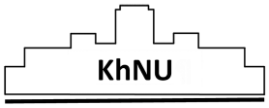
Reviewer: Vyacheslav Kalashnikov, Doctor of Sciences (Physics and Mathematics), Full Prof., Department of Systems and Industrial Engineering, Campus Monterrey, Monterrey, Mexico. E-mail: kalash@itesm.mx

Received: May 2019.

Authors:
Olga Melkozerova, Ph.D., Senior Lecturer, Department of Security of Information Systems and Technologies, V.N. Karazin Kharkiv National University, Ukraine.
E-mail: olja.mex@gmail.com
Oleksii Nariezhnii, Ph.D., Associate Professor, Department of Security of Information Systems and Technologies, V.N. Karazin Kharkiv National University, Ukraine.
E-mail: o.nariezhnii@karazin.ua
Serhii Malakhov, Ph.D., Senior Researcher, Department of Security of Information Systems and Technologies, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
E-mail: mailgate@meta.ua

Analysis of tools for automated software testing.
Annotation. Software quality testing (software) is a time-consuming and responsible stage for its development. This leads to a practical interest in automating basic testing procedures. As practice shows, the use of various software testing tools greatly facilitates this process. However, it is fundamentally important to use a particular tool, taking into account the specifics of each particular case. This circumstance is due to the large volumes of information being tested and the complexity of the operational documentation. The paper provides an overview and analysis of the capabilities of existing tools for automated software testing, with a description of the relevant technologies, purpose and scope. Examples of test preparation using Selenium, Serenity BDD and JMeter are given.

Keywords: Automated Testing; Automated Testing Tools; Automated Testing Technology.



Наукове видання

КОМП'ЮТЕРНІ НАУКИ ТА КІБЕРБЕЗПЕКА

Випуск 1(13) 2019

Міжнародний електронний науково-теоретичний журнал

Англійською, українською, російською мовами

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

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

V. N. Karazin Kharkiv National University Publishing

2019