УДК 004.621.3:519.816

S. Zybin

AN APPROACH OF PROVIDING THE REQUIRED LEVEL OF SECURITY THAT IS BASED ON THE SYSTEM ESTIMATIONS

This article reveals to the method of providing the required level of security that is based on the system estimation. The method was developed by using network models of Petri nets, Merlin, E-net.

Keywords: security program, protection, DSS, decision support system, information security, decision-making, simulating, Petri nets, Merlin nets.

Introduction

The creation of complex systems and the assessment of the quality of their functioning and performance provides of a wide range solutions for various tasks. Moreover, the intensive use of computer facilities and automation in them constantly adjusts views on the activities of these systems. The complex systems consist of decision support systems.

Improving the quality and reducing of the decision-making time when managing complex systems for various purposes are currently impossible without developing of the effective software and hardware. This problem is especially acute when decisions make in complex security systems (CSS), which work in on-line. The lack of time in such systems is especially felt. The aftereffects of untimely or incorrect of decision-making can be catastrophic.

Thus, the necessity for application of decision support systems (DSS) has appeared. The main task of the systems is help to specialists in the process of making-decisions in complex situations that arise during the operation of CSS. So the estimation of the decisions choice quality and their parameters should be based on models. Those models would allow estimating the application of the same system in different conditions of exploits.

Increasing the efficiency of CSS mathematical modeling can be provided by modeling both the complex system and the subsystems that are included in its. This necessity stimulates the development of models and algorithms that allow to solve the complex problems for a system management.

Therefore, the synthesis of CSS and DSS should be carried out in accordance with known criteria:

- the possibility of implementing any complex operation as a sequence of simpler actions;
- the modularity of construction;
- the main way of information exchange;
- the possibility of increasing computing power.

Development, analysis and research of mathematical models CSS and DSS, which are an integral part of information security programs, requires considerable time expenditures. The use of Petri Nets for such purposes can accelerate the process for solving similar tasks.

The main purpose of the article

The purpose of the article is to consider the possibility of using Petri nets for estimating the technical state of CSS and DSS, and estimating the quality of their functioning under various operating conditions.

The main part

DSS are an integral part of real-time CSS in terms of purpose, structure and functions. Therefore, the synthesis of DSS should be considered by taking into account the interaction of DSS algorithms with CSS functioning algorithms.

The most typical for modern control systems of CSS is the three-level structure of computing facilities.

There is a universal computing machine at the first level. It has a powerful potential for information processing. There are specialized computing devices at the second level. There are personal computers at the third that are part of automated workplaces.

The main functions in the formation of an information model and its management methods are the collection and processing of information. One of the important functions is the creation and issuance of control and information arrays to the control objects, as well as the adoption and use of the inputted information messages.

The computing devices execute the main function for the formation of an information model and its management. The most typical of these functions are the following [1]:

- the function for the formation of the information model;
- the management function of information model.

Implementation of decision-making support in CSS does not change the basic functions of computing means. It are associated with the formation of the information model. The simulation model allows you to estimate the efficiency of the system and eliminate conflict situations. In this case, the functioning of the system becomes situational.

We have to complete the system design process to estimate the efficiency of DSS consisting of CSS.

The main goals of the simulation are:

- the clarification of the technical decision choice of computer means and the division of functions between them;
- the verification of the functioning coherence for the technical DSS equipment's;
- the estimating of executive effectiveness for the CSS.

Below we can see the structural diagram of the simulation model in Fig. 1. This model performs the tasks that have described above.

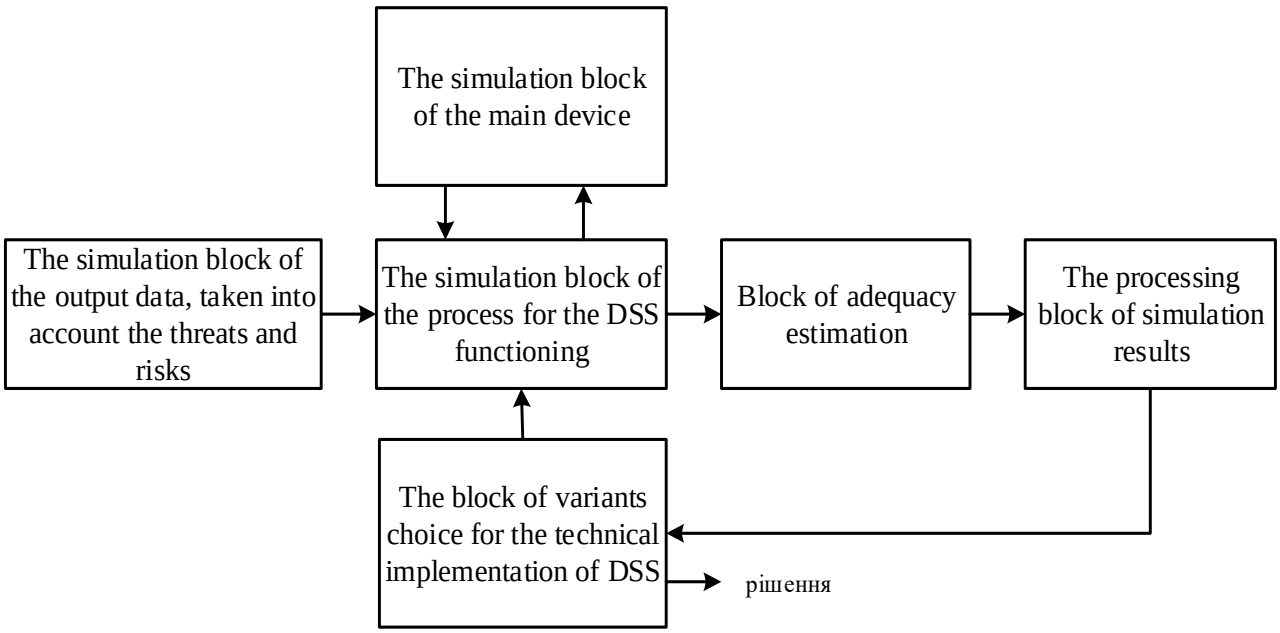

Fig. 1. Structural diagram of the simulation model at the system functioning

We need to take into account the actions of the intruders to estimate the effectiveness of DSS, taking into account threats and risks in the elimination of conflict situations.

The simulation block of the process for the DSS functioning is the main block. For the simulation of processes, those occur in the computing means, widely used: the queueing theory, probabilistic graphs, Petri net, etc. [1, 2].

The classic Petri net can be represented as follows – $P=(D, B, G, \Theta, M_0)$,

where B – is the final set of position $B=\{b_i\}$, $i = \overline{1,n}$;

D – is the final set of transitions $D=\{d_j\}$, $j = \overline{1,k}$;

$G:D\times B \rightarrow \{0,1\}$ – is the direct incidence function;

$\Theta:B\times D \rightarrow \{0,1\}$ – is the inverse incidence function;

M_0: $B \rightarrow Z$ – is the initial markup (marking) that defines the initial distribution of labels by network positions;

$Z=\{0,1,2,...\}$ – is the set of non-negative integers.

For each transition $d_j \in D$ can define a plurality of input $\Theta(t_j)$ and output $I(t_j)$ position:

$$\Theta(t_j)=\{b_i \in B/\Theta(b_i, d_j) = 1\},$$
$$I(t_j)=\{b_i \in B/G(d_j, b_i) = 1\},$$

Where is $i = \overline{1,n}$; $j = \overline{1,k}$.

The definitions of input set $I(b_i)$ and output set $\Theta(b_i)$ of conversion positions are inputted the similarly

$$I(b_j)=\{d_j \in D/G(d_j, b_i) = 1\},$$
$$\Theta(b_j)=\{d_j \in D/\Theta(b_i, d_j) = 1\}.$$

The marking of the network is a vector

$$M = \begin{vmatrix} m(b_1) \\ m(b_2) \\ \dots \\ m(b_n) \end{vmatrix},$$

where $m(b_i)$ – is the quantity of marks in the position b_i.

Petri net operates from marking to marking. The changing of the markup occurs because the one of the net's marking $d_j \in D$ is operated. The $\forall b_i \in B$ $[m(b_i) - \Theta(b_i,d_j) \geq 0]$ is a necessary condition for activation of one of the transitions d_j. The transition d_j for which the specified condition is fulfilled, is defined as being in readiness for operation or as an excited transition.

The transition operating d_j is changed the marking M_ξ to $M_{\xi+1}$, so marking d_j removes one label of each input positions and adds one label on each of the starting positions.

The main disadvantage of Petri classical nets is the lack of time in the dynamic describing of the functioning process for the system. We can eliminate this disadvantage by using two extensions of Petri nets: time nets and Merlin nets, which allow you to display the system's time settings [2].

The determining the time net - P_s is based on sets. We can use the help $P_s=(D, B, G, \Theta, M_0, J, \vartheta)$,

where is

$J=(\tau_1,\dots, \tau_i,\dots)$ – it is the growing sequence of real numbers (time base);

$\vartheta:B\times J \rightarrow$ J – it is the time delay function.

The fact of time is taken into account in this net by introducing a passive state of the label in a position. When the label arrives at the b_i position, it stays in the passive state at the time $\vartheta(b_i, \tau_s)$ and only then goes into active state.

The Merlin net is given by the expression $P_y=(D, B, G, \Theta, M_0, J^*, J^{**})$,

where $J^*=\{\tau_i^*\}$ – is the set of minimum delay times for the transitions $d_j \in D$;

$J^{**}=\{\tau_i^{**}\}$ – is the set of maximum delay times for the transitions $d_j \in D$.

The operation of any d_j junction for Merlin net may occur in a time not less than τ_i^* after its excitation and not more τ_i^{**}.

Further extensions of Petri Nets are knowns estimating nets or E-nets, which allow to see the dependence of processing processes on the type of tasks. These tasks are solved taking into account the priority of information processing [2]. However, they do not take into account one important feature of information processing. They do not take into account the probabilistic nature of information. Therefore, on the basis of time nets and E-nets as a device for simulating the simulation of the functioning of the DSS should be used modified Petri time nets, allowing to take into account the probabilistic nature of information processing.

Modified Petri time net is given by a set of sets $\widetilde{P_s} = \{D, B, G, \Theta, M_0, J, f(D_i)\}$,

where is

$B=\{B_{SP}, B_\tau, B_R, B_P\}$ – is a set of positions consisting of non-crossing B_{SP} subsets of ordinary positions;

B_τ – is the time positions;

B_R – is the control positions;

B_P – is the countable positions;

$D=\{D_{SP}, D_Y, D_F, D_R, D_P\}$ – is the set of basic transitions;

$J=\{\tau_i\}$ – is the staying time of the label in the position B_τ;

$f(D_i)$ – is the a function that determines the presence of a label in the control position B_R.

Epy five basic types of transitions are defined for the modified Petri net [3]. The logic of such transitions is defined by an indication of allowed markup changes. The operating of a D_{SP} type (normal transition) occurs when there is a label in the input position B_1 and the absence of a label in the starting position B_2, that is $(1,0) \xrightarrow{D_{sp}} (0,1)$.

We have $(1,0,0) \xrightarrow{D_F} (0,1,1)$ for the trasition (branching) D_F and the D_y association transition is described $(1,1,0) \xrightarrow{D_y} (0,0,1)$.

The control position of type D_R (branching on condition) is described as following:

$$(0,1,0,0) \rightarrow (0,0,1,0),$$
$$(1,1,0,0) \rightarrow (0,0,0,1).$$

The countable transition D_P is described of formula *(p, 1, 0)* $\rightarrow$ *(n-1,0,1)*, where is $n \geq 1$).

The following five major types of transitions allow us to simulate different situations encountered in processing information. The D_{SP} transition simulates an event that occurs when one condition is fulfilled. The D_Y transition is used if two or more conditions are present. The branching of the information flow is displayed by the D_F transition. Transition type D_R is used if it is necessary to change the direction of information flow under some conditions. The T_P transition is used when a counter is arranged.

There are several types of positions in the modified Petri net:

- B_{SP} – is the usual position, the mark will have removed immediately after the permission of the original allowed;
- B_τ – is the time position, the mark will have removed after time τ finished;
- B_R – is the control position, the mark appears based on the result of the function calculation $f(b_i)$ і керуюча позиція, у якій мітка з'являється по результату обчислення функції $f(b_i)$ and disappears when the output is engaged;
- B_P – is the count position, the quantity of marks is determined by the counter of the cycle for the execution of the program the number of labels which is determined by the counter of the cycle of execution of parts of the program.

The operation of the modified Petri net is a sequential execution of three phases [3].

The phase of pseudo-readiness presents at all transitions. During this phase, there is a check of the transitions to the permissibility. It is means there is a presence of at least one mark in all

incoming positions. The exception is the D_R control transition, which has enough labels in the same position.

The transition enters the readiness phase if it is allowed. The determining of the time result of finding the marks, during this phase executes in the incoming time positions. The transition enters the active phase after the end of this time. If the transition is B_{SP}, then it enters the active phase immediately.

The markup changes in the active phase according to the transition control.

Thus, the apparatus of the modified Petri net allows us to construct a sufficiently complete model of the algorithms functioning that reflect their structure, logic of work and time characteristics.

For the effective use of a wide spectrum of capabilities hardware Petri nets, it is necessary to create Petri based hardware nets, a system of special mathematical support with a set of means for describing, inputting, translating, compiling, debugging, simulating the model, processing the results of modeling and analysis.

When constructing a simulation system of Petri nets, a significant role is played the choice:

- description of source models;;
- of internal representation method of the described model and on its basis the organization of the of modeling algorithm.

The inner-system representation of Petri nets can be organized in the form of arrays or in the form of list structures.

In the presence of DSS in CSS, the inner-system representation in the matrix form of Petri net can be described by two arrays: the incidence array E of the dimension $p \times d$, where p is the number of vertices of the places, d is the number of vertices of the model's transitions, and the array of the F markings of the dimension, which are determined as follows [4]:

1) $E(i,j)=1$, if $B_i \in B_{t_j}^I$; $E(i,j) = 0$, if $B_i \notin B_{t_j}^I$;

2) $F(i,j) = \alpha+\beta$, where $\alpha = 1$, if $B_i \in B_{t_j}^I$;

$\alpha = 0$, if $B_i \notin B_{t_j}^I$;

$\beta = -1$, if $B_i \in O_{t_j}^o$;

$\beta = 0$, if $B_i \notin O_{t_j}^o$.

Let A^j – is j column of array A.

Then you can certify:

а) t_j transition can be executed, if $E^j - d_0^{-(k)}$;

б) the next markup after the operation t is calculated by the formulas

$$d_0^{-(k+1)} = d_0^{-(k)} + F^{(j)}$$

$$-\left[E^j \to d_0^{-(k)}\right] \equiv \left[E^j d_0^{-(k)}\right] = \left[E^j / d_0^{-(k)} = 0\right].$$

Consequently, the condition for running the transition t_j is to fulfill the condition $E^j d_0^{-(k)} = 0$, and the subsequent mark is calculated as follows:

$$d_0^{-(k+1)} = d_0^{-(k)} \oplus C^j,$$

where

$\oplus$ – is denotes the operation EXCLUSIVE OR;

$C(i,j) = 1$, if $F(i,j) \neq 0$;

C(i,j) = 0, if *F(i,j) = 0*.

Here all operations are performed over the vectors of the Boolean variables, which allows you to implement this method quite efficiently with the use of computer technology.

We introduce the representation of each t_v tv transitions in one of the places $B_\varepsilon^\tau \in B_{t_v}^1$ to increase the speed. To start the transition t_v it is necessary (but not enough) to fulfill the condition $d(B_\varepsilon^\tau) = 1$.

We define the vector of the Boolean variables *R* of dimension $d \times 1$, as well as the array *A* and *W* with dimension $d \times d$:

- *R(j) = 1*, if $d(B_i^\tau) = 1, B_i^\tau \in B_{t_j}^1$;
- *W(i,j) = 1*, if t_j and t_i are represented by same place B_i^τ;
- *A(i,j) = 1*, if t_j is represented by place $B_i^\tau \in \Theta_{t_j}$.

Then, after triggering t_j, the next markup is calculated by the formula $R^+ = R \oplus A^j \oplus W^j$ and is modeled by the algorithm. In addition, $R^+ = R \oplus L^j$.

Here, $L^j = A^j \oplus W^j$ allows you to save the amount of memory. With this approach, you can shorten the execution time of the program while reducing the amount of memory. This is due to the array L_i vector R. To reduce the amount of memory it is expedient to use the inner-system representation of models in the form of stack structures, because E, F, L are the sparse array. As a result, the memory size is linearly dependents on the values of d and p, whereas in the case of an array representation, this dimension is proportional to d × p.

One way to achieve a compromise between the complexity and probability of a mathematical model is to simplify the equivalent of a net object. Simplification takes place through the operation of the system [5, 6] on the basis of the apparatus of fuzzy relations in space, which is due to the base of CSS and DSS. This base can be expanded. The same database records the behavior of the system in the presence of risks, threats and external influences. Either models obtained in this way have a controllable dimension and are transformed into a compact or an expanded form based on rigorous mathematical rules. The probability of the CSS and DSS model is not the output, but the input parameter for modeling. Hence, the main advantage of such an approach follows. The route model, with the probability that it is predefined, allows you to predict the dynamics of developments around the information security program in the presence of threats and risks, as well as the state.

Let's consider in more detail the principles of constructing routes, route models and modeling information environment. We assume for *X* the universal set of possible relations of the simulated object. Let *X* be modeled with the required probability φ by a set of descriptions N_0 consisting of elements $\bar{n}$.

So:

$$N_0 \leq x;$$

$$N_0 = \{N/\bar{N} \in X, \mu(\bar{N}) \geq 1 - \varphi\}, \quad (1)$$

where $\mu(N)$ – is the property function of the description $\bar{N}$ by the set *X*.

The route is a set of level $\alpha \neq 1 - \varphi$, as a reflection of the Markov process with fuzzy initial conditions according to the fuzzy set of descriptions N_0;

$$N = \{\bar{N}/N_0, \mu(\bar{N}) > \alpha\} \quad (2)$$

However, taking into account rules for organizing elements in N_0, the route can be represented as $A\bar{B}P = (B, D, KS)$, where N_0 reflects the character of the *ABP* component.

We will assume that the set of relations corresponding to the "normal" route of the N_H is defined as:

$$N_H = \{\overline{N}/\overline{N} \in N_0, \mu_H(\overline{N}) > \beta\}, \quad (3)$$

where β – is the parameter that defines the stability CSS to external influences.
At the same time, the following statement is truth for the "experimental" route

$$N_{\mathrm{E}} = \{N/\overline{N} \in N_0, \mu_{\mathrm{E}}(\overline{N}) > \beta^1\}, \quad (4)$$

where β^1 – is the given parameter of the boundary instability of CSS.
When expanding and narrowing the sets of modeling relations, the following principles should be guided:

- extension of the normal route taking into account the experimental route розширення

$$N_1 = \{\overline{N}/N \in N_0, N_1(\overline{N})\}, \quad (5)$$

where $N_1(\overline{N}) = \begin{cases} 0, \ if \left[\mu_E(\overline{N}) X_{\mu_E}(\overline{N})\right] < \beta \\ \max[\mu_E(\overline{N})], \ if \left[\mu_E(\overline{N}) V_{\mu_H}(\overline{N})\right] \geq \beta \end{cases}$

- narrowing the experimental route taking into account the normal route

$$M_2 = \{\overline{N}/N \in \overline{N_0}, N_2(\overline{N})\}, \quad (6)$$

where $N_2(\overline{N}) = \begin{cases} 0, \ if \left[N_E(\overline{N}) V_{\mu_H}(N)\right] \geq \beta \\ \max[N_E(N), N_H(N)], \ if \ [N_E(N) N_H(N)] < \beta \end{cases}$

It follows from conditions (5) and (6)

$$\lim_{\beta \to 0} N_1 = \lim_{\beta \to 0} N_2 = N_0 \quad (7)$$

The speed of conversions and the probability of placement for the positions of the modeling Petri net is a measure of informativeness of their respective relations.

For $\beta = 1$ in Petri net, which is synthesized on route sets, enters the most "live" transitions of Petri nets constructed on N_0 [7]. As the number of nodes in Petri net grows, the function of the transition of the set of "live" transitions decreases. We replace the concept of speed with the expert evaluation of the membership transition of the "live" set transitions. This succeeds in moving away from the direct solution of the possibility question for the operation of a transition.

Let's denote the initial state through N_p, for sets states of type as a route sets, and achievable from it as N_p^+. Then the prediction as a linear operator is described as follows:

$$F = N_p^- = N_p^+, \quad (8)$$

where F – is linear operator of prediction:

$$N_p^- \underline{W} \; and \; N_p^+ \underline{W} M_1.$$

The prediction as a functional is determined in the basis of N_0 as a function of belonging to the state N_p^- - the estimations set of the CSS technical condition [8]. Aspects of the predictions have their predictions in the *AVR* and are formalized as a linear operator in the space, which is generated by N_0 and as a functionality due to the linear form in the space N_0.

From the relation (8) it follows that the forecast as a linear operator and as a functional forms the tree of possibilities, because according to the definition of expressions (5) and (6) it follows that the power N_1 is greater than N_2. With machine realization, this leads to the solution of the combinatorial type problems and to the exponential growth of the model dimensions. As a result, we perform clipping of the branches, that is, we only consider those branches of the tree possibilities whose function is N_0 not less than β. The basis for implementing the reduced approach to PC is the allocation and analysis of the called stationary state of the system of technical protection. For N_0, the set of stationary states is defined as

$$N_0 \leq N_W$$

$$N_W\{\overline{N}/N \in N^0, N^W(\overline{N})\} \cong 1,$$

where N_W – is a set of stationary states.

All elements of the N_W are roots of the normal route in the absence of external influences. External influences form a space of perturbations, the basis of which is the elementary influence [8]. Each element of the N_W corresponds to an fuzzy limited subspace of the disturbance space. In other words, the elements of N_W are assigned sensitivity to the basic elements of the disturbance space. Thereby is given rise to the elements of the sets. The set of states of each stationary state originates from a plurality of elements, one for each zero element of the subspace basis of perturbations. Relationships between sets and a set of stationary states of a field is

$$N_Э \cap N_W = N_H \cap N_W = N_W.$$

In other words, the basic effects generate characters trees capabilities.

An analysis of the stationary CSS states should identify the relationship between them. DSS is used in the case of a large complexity of the system. Expert assessments determine the interconnection of elements N_0. The result of the analysis is the normal path of stationary states. It is the basis for constructing a probability tree and predicting the technical state of CSS and the level of protection.

DSS as a part of the information security program includes the nodal moments of the CSS functioning. Therefore, it reflects the behavior of elements and subsystems according to the algorithm. Thus, DSS solutions are a model of the standard work of the information security program. The prediction of the technical state of such a program is based on the Markov character of the subsystems functioning, on the one hand, and on the system of assessments and recommendations of the DSS from the other side.

To determine correctly of the functioning program quality of information security with the advice and DSS is necessary to satisfy the following requirements [5,6]:

- the system of the technical state assessments and the quality of operation should include the priorities of the corresponding source branches of the stationary states for the Petri net. They are expressed in the form of belonging functions to states of the output branch for set of technical states and the operation conditions of the information security program;

- the research depth, the details of the technical states and the quality (conditions) of the operation of the information security program operation is determined by the given probability φ .

Taking into account these requirements, the model is implemented on the basis of expressions (1) ÷ (8). It is a model based on associative principles. It can vary widely, depending on the required probability of modeling the depth of search in the database and connecting nodes of the Petri net. Because the data in the database is organized in the form of a set of intersecting trees. Crossing trees should be understood as a fuzzy relation [9]. The intersection node is a fuzzy set, which is assigned a measure in the form of a membership function of the node tree association. Depending on the transitional requirements of the association, they may expand, divide or form new, wider associations with other associations. Summary of routes in a database organized associative access to specific classes of information security programs. They are supplemented simultaneously with new information contained in the database. Old information is deleted.

Proceeding from this, the means of intellectualizing decision-making processes are the most important and practically necessary element in the field of information technology at present. They are the basis of information security.

References

1. Герасимов Б.М. Системы поддержки принятия решений: проектирование, применение, оценка эффективности / Герасимов Б.М., Дивизинюк М. М., Субач И. Ю. – Севастополь: НИЦ ВС Украины "Государственный океанариум". – 2004. – 320с.

2. Питерсон Дж. Теория сетей Петри и моделирующие системы / Питерсон Дж. – М.: Мир, 1984. – 264с.

3. Котов В.Е. Сети Петри / Котов В.Е. – М.: Наука, 1984. – 160с.

4. Капустян М.В. Применение сетей Петри для оценки технического состояния систем защиты информации /Капустян М.В., Хорошко В.А., Чирков Д.В. // Сучасний захист інформації, № 1, 2011. – С. 10-15.

5. Zybin S. The one method to decision making support for formation of complex security information programs. // Сучасний захист інформації: наук.-техн. журн. / Держ. ун-т телекомунікацій. – Київ: Вид-во ДУТ, 2016, № 4, С. 73 – 79.

6. Тискина Е.О. Проектирование систем защиты информации и систем поддержки принятия решений для них / Тискина Е.О., Хорошко В.А. // Сучасний захист інформації, Спецвипуск, 2010. – С.25-31.

7. Моржов С.В. Применение сетей Петри для моделирования параллельных процессов / Моржов С.В., Хорошко В.А. // Проблемы управления и информатика, № 2, 2004. – С.86-94.

8. Кобозева А.А. Анализ информационной безопасности / Кобозева А.А., Хорошко В.А. – К.: Изд. ГУИКТ, 2009. – 215с.

Майника Э. Алгоритмы оптимизации на сетях и графах / Майника Э. – М.: Мир, 1981. – 323с.

Надійшла 19.06.2017 р. Рецензент: д.т.н., проф. Горбенко І.Д.