Date of publication JUN-30, 2023, date of current version MAY-26, 2023.
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ ijc.22.2.3082

Distributed Discrete Malware Detection Systems Based on Partial Centralization and Self-Organization

SERGII LYSENKO[1], BOHDAN SAVENKO[2]
[1]Khmelnytskyi National University, Khmelnytskyi, Ukraine

Correspondent author Sergii Lysenko (e-mail: sirogyk@ukr.net).

ABSTRACT Malware detection remains an urgent task today. Various means for the development of information technology and providing users with useful applications are being transformed by attackers into tools for malicious influences and manifestations. A variety of countermeasures and detection tools have been developed to detect malware, but the problem of malware distribution remains relevant. It is especially important for enterprises and organizations. Their corporate networks and resources are becoming objects of interest to intruders. To counteract and prevent the effects of malware, they have various systems in place. In order to improve the counteraction to malicious influences and manifestations, the paper proposes the use of distributed discrete systems, in the architecture of which the principles of self-organization, adaptability and partial centralization are synthesized. Such tools and their functioning will be difficult to understand for attackers and, therefore, will be difficult to circumvent. The architecture of the proposed tools will integrate the implemented methods of malware detection for a holistic counteraction to malware. Such a system will be a single sensor that will detect malicious influences and anomalies. To organize its functioning, descriptions of characteristic indicators are needed. The paper presents the developed mathematical models for determining the values of characteristic indicators. According to obtained values the system architecture was formed. In order to evaluate the sustainability of the developed distributed discrete system a set of experiments were conducted. In addition, to study the accuracy of malware detection, the developed system was tested for the possibility of worm virus detection. Experimental studies have confirmed the effectiveness of the proposed solution, which makes it possible to use the obtained solutions for the development of such systems.

KEYWORDS distributed systems; discrete systems; malware detection; principle of partial centralization; self-organization; cybersecurity; cyber threats; malware detection

I. INTRODUCTION

Malware detection systems in computer networks, as well as in their hosts, should be based not only on current known detection methods but also implemented in such tool architectures that would involve their elements in improving detection in con-junction with detection methods [1]. The response of malware (malicious software) detection systems in computer networks by dynamically restructuring their architectures in the face of malicious influences and anomalous manifestations creates additional obstacles for attackers and malware. Such a dynamic architecture restructuring should be coordinated and aligned with the use of detection methods [2]. Creating obstacles for attackers and malware to understand the functioning and behavior of detection tools at the architectural level provides an advantage to users of computer systems and networks. Achieving the advantage requires, in addition to methods that are focused directly on the detection of malware, to provide components or elements in the detection tools that change the architecture of the detection system in conjunction with the detection methods, but are not focused on the detection of specific types of mal-ware [3]. Such components or elements must ensure the functioning of the malware detection system without the intervention of the system administrator or user in making decisions on further functioning under the influence of malware and not be predictable in their further actions for attackers and users.

An architecture that allows for self-organizing and adaptive systems is a promising solution for such tools. Self-organization allows the system to determine its next steps without the need for user or administrator intervention [4-6]. Adaptability al-lows the system to dynamically rebuild its architecture depending on the state of the system, external events in computer systems and networks in which it is installed.

This paper is organized as follows. Section 2 presents the state-of-art. Section 3 describes the architecture of partially centralized malware detection systems. Section 4 discusses the component architecture of partially centralized malware detection systems. Section 5 proposes the experiments and the

efficiency of the proposed approach. Finally, we present our conclusions and future research.

II. THE STATE-OF-ART

Let's examine the concept of distributed systems and their design principles [1-3] for efficient utilization in computer networks, including those designed for malware detection. Publications [4, 5] suggest approaches to tackle the challenge of con-structing a directed minimum spanning tree for use in building distributed systems. Paper [6] demonstrates a method to con-struct an overlay network with a constant degree and a specified diameter within a reasonable timeframe, starting from any initially low-connected graph.

Mathematical aspects for the organization of self-organized dynamic systems are presented in [7]. It is proposed to de-scribe the organization of such systems through the characteristics of the processes that will occur in the system or the environment it controls.

A distributed service-oriented multiagent system is presented in [8]. Agents must cooperate with each other to perform decentralized service discovery tasks. The structure of the system affects the efficiency of service discovery. Therefore, it is necessary to use a structural self-organization mechanism to facilitate decentralized service discovery in the system.

Paper [9] presents the importance of self-organization in solving the problem of overcoming complexity. The developed approach to the implementation of self-organization is important for its use in technical systems as well.

Paper [10] presents a study of higher-order dynamic systems and dynamic topology. As a result, an analysis of the relationship between higher-order interactions and collective behavior is proposed.

Modeling in [11] shows that gradual changes in the system can lead to a state of self-organized criticality. The process, when approaching this state, may encounter changes that may cause nonlinear bursts of process complexity. Digital technology can be developed and used to influence the probability and severity of these transformational phase changes.

In [12], it is proved that at the limit of neural networks, the system oscillates around a critical point. This is important in the context of its sustainability and the forecast of further steps. Paper [13] analyzes self-organized quasi-criticality, which works by the spontaneous emergence of universal facts. As a result, achieving this state allows the system to decide on the number of components in it.

In [14], the aim of the study was to develop and test a new concept for modeling a distributed system and managing a dis-tributed system for the implementation of dis-tributed production nodes.

In [15], for the effective functioning of cyber-physical systems, a communication system between them was developed ac-cording to their model.

In [16, 17], based on the principles defined by the software, a holistic architecture for cyber-physical systems and IoT applications is proposed. The issues of scalability, flexibility, reliability, interoperability, and cybersecurity are addressed. The architecture uses computing units owned by intelligent agents that can be used for decentralized control and data processing in the network. In addition, a middleware layer is proposed to encapsulate devices and services for time-critical operations in highly dynamic environments. Also, many vulnerabilities to cyberattacks are identified and software-defined solutions for such systems are integrated.

A new class of self-organized cyber-physical systems was obtained in [18] by combining cyber-physical systems and self-organized computing. Thus, the creation of a system with significantly increased controlled autonomy, which is a basic requirement for many new and future applications and technologies. Self-organized cyber-physical systems are located in a physical environment and are limited in their resources. They understand their own state and environment. Based on this under-standing, they are able to make independent decisions during execution. The authors have identified five key challenges for future self-organized cyber-physical systems.

Works [19, 20] analyze autonomous agents that are able to work on a goal and interact with the environment and other systems independently. At the same time, they must understand their environment, their goals, tasks, and other agents that are near-by.

Paper [21] formalizes processes for collective behavior observed in living systems. They can be localized in space and are not limited by geometric constraints. In the context of describing complex systems, this work reveals approaches to the formation of self-organized systems.

The organization of complex systems is described in the monograph [22]. It describes the components of information, how to build them, and what difficulties may arise.

Paper [23] discusses self-organization as a general mechanism for creating a new structural model of systems. Characteristics of self-organizing behavior, such as open-ness, nonlinearity, internal randomness, internal feedback, information network, and holographic construction, provide appropriate conditions and a basis for the self-organizing evolution of the system from the aspects of the information function of the environment, the maintenance and construction of the general information basis of the system, and the study of the new information mode of the system.

Let's explore distributed systems for detecting malware, specifically focusing on Network-based Intrusion Prevention (NIPS) [24], which monitors network traffic and blocks suspicious data flows; Intrusion Prevention Systems (IPS) for wireless networks (Wireless Intrusion Prevention Systems, WIPS) [25], which monitors activity in wire-less networks; and network behavior analysis systems (Network Behavior Analysis, NBA) [26], which analyze network traffic to identify unusual patterns. Methods and techniques for building distributed systems and detecting malware are discussed in scientific literature [27-30].

When it comes to malware, let's consider worms as an example. Studies [31-32] have decomposed viruses and worms based on their main functional components, providing a catalog of six functions performed by these malicious programs and classifying different implementation methods. This catalog and classification serve as a foundation for enhancing current reactive virus detection technologies and developing new proactive approaches.

A paper [33] analyzes a type of Trojan horse that can modify other computer pro-grams during execution, such as by copying itself into them. Another study [34] examines malicious programs for signs of viruses, worms, Trojans, and rootkits, and

pro-poses specific countermeasures for their recognition. In [35], a SIQR model is con-structed to study the spread of Internet worm viruses using a two-factor model and mathematical equations to analyze the dynamic properties and spread patterns. This model provides a theoretical basis for controlling and predicting Internet worms.

In [36], the authors assume the existence of multi-vector worms and identify a couple of them through traces of an attack collected on a honeypot. These worms utilize multiple attack methods to spread, but only employ one method against a specific target, making them indistinguishable from classical worms that always use the same attack vector or sequence. Lastly, [37] analyzes various types of malicious software, including viruses, worms, Trojans, spyware, keyloggers, botnets, rootkits, ransomware, scareware, and random downloads. Different aspects of the distributed systems' usage are presented in [38-44].

Thus, the properties of self-organization in complex systems are important and can be realized. Modern networked dynamic systems have some features that allow to realize the benefits of self-organization.

III. ARCHITECTURE OF PARTIALLY CENTRALIZED MALWARE DETECTION SYSTEMS

An important decision in the design of malware detection systems is the decision on the system's decision center to determine its further functioning. Such a system decision center can be located in one place, i.e., be a single one, or in several, i.e., be dis-tributed. Also, it can be the same in each component of the system, in which case the entire system will be decentralized. When designing malware detection systems, it is necessary to take into account that attackers or malware will definitely try to interfere with the operation of such specialized systems and, therefore, will investigate the location of the system center. The use of decentralized architectures for malware detection systems is effective, but the time for decision-making in them is longer than for systems developed using centralized architectures, which is an essential characteristic of efficiency in modern process conditions. In addition, research into the detection of malware in individual components of decentralized systems can be ongoing and this will affect a significant number of system components that are free of malware. This will generally load the computing resources of all computer stations in the network. Therefore, the choice of architecture for a malware detection system will be made from centralized and hybrid architectures. For a large malware detection system distributed in a computer network, the presence of a single center creates a problem, because the failure of it or the computer system in which it is installed will lead to the loss of the system as a whole and the system components installed in the other network computer stations will lose their effectiveness in detecting malware. Therefore, it is advisable to consider a centralized architecture with partial centralization, which should be consistent and combined with the self-organization and adaptability of the system as a whole. Partial centralization will be considered as a kind of the hybrid architecture, in which the decision-making center in the system will be distributed among a small number of components and, if necessary, will migrate completely between them. This architectural organization of the system's decision center will make it difficult for attackers and malware to find it. The proposed partially centralized architecture will also contain fewer connections compared to the decentralized architecture, and more connections compared to the centralized architecture, approximately the number of components allocated to move the center of the system. Maintaining connections between system components requires resources and especially time. If certain connections are lost, time will also be spent on clarifying or reconfirming the connections, which will affect the efficiency of the system.

Partial centralization, which is proposed to be implemented in the system architecture will provide a dynamic restructuring of the system from decentralization, which includes partial centralization, in order to increase the centralization if needed. For example, when removing some of the components that do not contain the center of the decision-making system, which will increase the ratio of the number of components with the center of the system in relation to the number of components without a center. In addition, the system's decision-making center may decide to remove some of the connections between system components if necessary. Therefore, the level of centralization in the system can change dynamically depending on the state of the system as a whole at a certain point in time and the functioning processes in computer systems and the network.

Thus, the basis of the malware detection system will be an architecture that will synthesize partial centralization, self-organization, and adaptability. This will make it possible to create malware detection systems of this class that do not depend on the user or administrator to make decisions about their further work or next steps, that can be rebuilt dynamically and that will contain several components with a system decision center to improve the system's overall resistance to malware. In addition, these inherent capabilities in the system architecture will be aligned with specific malware detection methods and will be involved in the detection process from the system architecture level.

Let S be a partially centralized system, then let define it according to the formula according to the components, taking into account the distribution in the nodes of the computer network as follows:

$$S = \bigcup_{i=1}^{N} S_i, \quad (1)$$

where S_i is a i-th system component; N – the number of components in the system that are installed in computer stations in the network.

Some components of partially centralized systems will contain the center of the system or parts of it. Also, in certain components, the center can be moved. The rest of the system components will not contain the system center, but will have this option if necessary. Thus, it is not necessary for all system components to contain the system center at the same time. Otherwise, if all components of the system contain the system center at the current time, it degenerates into a decentralized system. If the center of the system is in one component at a certain point in time, it becomes centralized. Thus, the system center can either be in one component and move, or it can be distributed among several components if needed. Also, based on such variants of the architectural features of the center, the entire system can be partially centralized, but in extreme cases it can degenerate into a centralized or decentralized system.

Then, let us define a partially centralized system S with two types of its components as follows:

$$S = S_1 \cup S_2 \cup \ldots \cup S_k \cup \ldots \cup S_N, \quad (2)$$

where S_k is a k-th system component; N – number of components in the system that are installed in computer stations in the network; components from 1 to k may contain the center of the system; components from $k+1$ to N do not contain the center of the system.

If $k = N$, then the system becomes decentralized. If $k = 1$, then the system becomes centralized. If in a partially centralized system the center of the system is in one component, then it becomes centralized, and if the center of the system moves between several components or is distributed among several components, then such a system becomes k – centralized and contains more additional connections between components for the center. Therefore, for the designed system, the choice of a partially centralized architecture will be based on $1 < k < N$. Let us represent the system as a graph (Fig. 1) according to formula (2.2), taking into account the horizontal connections between the components. The figure shows three graphs of possible architectures for the designed system. This variant of the architecture, as proposed in Fig. 1 has the versatility that allows you to move from it to other architectures. This will allow you to flexibly adapt the system to the current tasks that will arise in the computer network when attacked by intruders or malware.

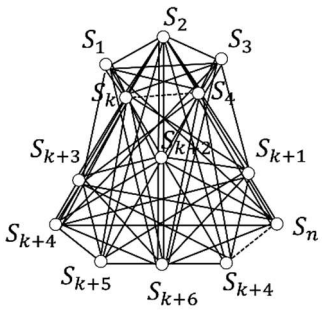

Figure 1. Partially centralized system architectures.

Then, the architecture model M_S of a partially centralized system S is defined according to its components and the relationships between them as follows:

$$M_S = \langle S, G_S \rangle, \quad (3)$$

where G_S– is the graph from Fig. 1, which reflects the connections between the components of a partially centralized system S.

Taking into account the division of the system components into two subsets according to the criterion of the presence of a center in them and without it, we obtain a refined model of the architecture $M_{S,k}$ of a partially centralized system S according to the formula:

$$M_{S,k} = \langle \langle S_1, S_2, \ldots, S_k \rangle, \langle S_{k+1}, S_{k+2}, \ldots, S_N \rangle, G_S \rangle, \quad (4)$$

where G_S – is the graph from Fig. 1, which reflects the connections between the components of a partially centralized system S; k is the number of system components that can have a system decision-making center; $S = S_1 \cup S_2 \cup \ldots \cup S_k \cup \ldots \cup S_N$; N is the number of components in the system that are installed in computer stations in the network.

In addition to the distribution of system components, which are specified by subsets and, accordingly, vertices, three types of connections are distinguished in the architecture model $M_{S,k}$ of a partially centralized system:

1. components with the center of the system;
2. components without a system center
3. components with a center and without a center between them.

In this case, the graph G_S is complete, i.e., the connections between the system components are available to all of them. However, for more efficient operation and hiding the system's capabilities, connections between system components can be specified by different trees of the graph G_S and, thus, their number will decrease, as well as hide them from an attacker or malware expected messages. The determination of the variants of the trees of graph G_S will be set by the decision-making center of the system.

The definition of the architecture of a partially centralized system S by its architecture model given by formula (4) meets the requirements for the possibility of dynamic configuration change, separation of the decision-making center, distribution of components by capabilities with the presence of a decision-making center in them, and the depicted architecture from Fig. 1 in the form of a graph G_S. Therefore, it is the basis for further synthesis of the properties of adaptability and self-organization in it, the implementation of which will be carried out directly in the system components, mainly those in which the decision-making center of the system will be located. That is, for a system to be able to self-organize and adapt depending on the environment and processes in computer systems and networks, it is necessary to synthesize these properties in the center of the system's decision-making.

IV. COMPONENT ARCHITECTURE OF PARTIALLY CENTRALIZED MALWARE DETECTION SYSTEMS

The synthesis of adaptability requirements and their self-organization in partially centralized malware detection systems requires its implementation in the decision center as part of it. Also, there may be several decision centers in the system and they will coordinate further steps of the system. In this case, these requirements will be separate in each of the components containing the system center. Since the decision center of the system is located in the system components, let's consider the architecture of the system components.

Components of partially centralized systems S according to the graph in Fig.1 are divided into two types. Let's consider their architectures separately and determine the possible presence of a system decision center in them, taking into account their purpose in the system. We will form the architecture of the system component in general for two types of components, since these components will have the same functions, since they are components of the same system, and will differ only in the additional presence of the center function. Components that contain the decision-making center of the system may not be the ones in which decisions are made at certain periods of time, but perform the functions of components without a decision-making center. The component includes a function to ensure the functioning of the component in a separate node of a computer network, and a function to establish different variants of connections between different types of system components, a function to process external messages from the rest of the

components of a partially centralized distributed system S, a function to make decisions in the system component regarding further steps of the system functioning, a function to detect malicious software, a function to ensure decision-making in the system S, a function to coordinate the decision with the rest of the components with the system center, a function to notify all components of the system S about the next steps. These component functions will also contain functions for performing the main assignment tasks and auxiliary support tasks. Thus, they will be specified as sets of functions for a specific purpose and divided into subsets of functions for performing specific tasks. Therefore, the functions of a component of a partially centralized system S contains the set functions. Let us define a component of a partially centralized system S by a list of its set functions according to the formula:

$$\Psi_{S_i} = \bigcup_{j=1}^{N_{S_i}} \Psi_{S_i,j}, \quad (5)$$

where Ψ_{S_i} is a set of functions S_i of components of a partially centralized system S; i is the number of the system component; $i = 1, \dots, N$; N is the number of components in the system that are installed in computer stations in the network; $\Psi_{S_i,j}$– is j-th function that ensures the execution of one functional task in the i component; $j = 1,2, \dots, N_{S_i}$; N_{S_i} is the number of functions in the S_i component that are combined in the set function Ψ_{S_i}.

In particular, let $\Psi_{S_i,1}$ be a function for ensuring the functioning of a component in a separate node of a computer network, $\Psi_{S_i,2}$ be a function for establishing different variants of connections between different types of system components, $\Psi_{S_i,3}$ be a function for processing external messages from the rest of the components of a partially centralized distributed system S, $\Psi_{S_i,4}$ be a function for making decisions in a system component regarding further steps in the system functioning, $\Psi_{S_i,5}$ be a function for detecting malicious software, $\Psi_{S_i,6}$ be a function for ensuring decision-making in the system S, $\Psi_{S_i,7}$ be a function for coordinating $\Psi_{S_i,8}$ is a function for notifying all components of the system S concerning the next steps and other functions that extend the capabilities of the system component in a computer network node.

Let's divide functions into two types. The first type includes those that contain all the functions that are provided for system components and contain the means of the decision-making center in the component. The second type includes those that contain all the functions, except for those responsible for the formation of the system's decision-making center in the component.

Let us define the matrix as a refined model of the architecture $M_{S,k,\Psi}$ of a partially centralized system S according to the functions in the system components, taking into account the formulas (4):

$$M_{S,k,\Psi} = \begin{pmatrix} \Psi_{S_1,1} & \dots & \Psi_{S_k,1} & \dots & \Psi_{S_N,1} \\ \Psi_{S_1,2} & \dots & \Psi_{S_k,2} & \dots & \Psi_{S_N,2} \\ \dots & \dots & \dots & \dots & \dots \\ \Psi_{S_1,N_{S_{max}}} & \dots & \Psi_{S_k,N_{S_{max}}} & \dots & \Psi_{S_N,N_{S_{max}}} \end{pmatrix}, (6)$$

where $\Psi_{S_k,j}$ – is j-th function that is part of the set function Ψ_{S_k} of the component S_k; $k = 1,2, \dots, N$; N – the number of components in the system that are installed in computer stations in the network; k – number of system components in which the system center can be located; $S = S_1 \cup S_2 \cup \dots \cup S_k \cup \dots \cup S_N$; $j = 1,2, \dots, N_{S_{max}}$; $N_{S_{max}}$ – is the largest number of functions that can be set in the component S_k.

Some of the functions developed for components may not be installed. That is, a system component may have fewer functions. In particular, for example, there may be no functions that ensure decision-making by the system, i.e., the component does not have a decision-making center or part of it. The absence of a function in a system component at a particular node in the network will be denoted by a null function.

We introduce a binary matrix that indicates the existence of functions used to compose the system components, based on the matrix obtained from formula (6). The absence of functions in the components is set to {0}, the presence - {1}. Then, the bit matrix is defined as follows:

$$P(M|S,k,\Psi) = \begin{pmatrix} P(\Psi_{S_1,1}) & \dots & P(\Psi_{S_k,1}) & \dots & P(\Psi_{S_N,1}) \\ P(\Psi_{S_1,2}) & \dots & P(\Psi_{S_k,2}) & \dots & P(\Psi_{S_N,2}) \\ \dots & \dots & \dots & \dots & \dots \\ P(\Psi_{S_1,N_{S_{max}}}) & \dots & P(\Psi_{S_k,N_{S_{max}}}) & \dots & P(\Psi_{S_N,N_{S_{max}}}) \end{pmatrix}, \quad (7)$$

where

$$P(\Psi_{S_k,j}) = \begin{cases} 0, if\ function\ \Psi_{S_k,j}\ is\ missing\ in\ the\ component; \\ 1, if\ function\ \Psi_{S_k,j}\ is\ present\ in\ the\ component. \end{cases}$$

$\Psi_{S_k,j}$ – is a j-th a function that is part of the set function Ψ_{S_k} of the component S_k; $k = 1,2, \dots, N$; N – the number of components in the system that are installed in computer stations in the network; k – number of system components in which the system center can be located; $S = S_1 \cup S_2 \cup \dots \cup S_k \cup \dots \cup S_N$; $j = 1,2, \dots, N_{S_{max}}$; $N_{S_{max}}$ – is the largest number of functions that can be installed in component S_k.

Also, for the matrix given by formula (6), let's introduce a matrix that will reflect the activity of the functions at the current time and their availability, as follows:

$$P_t(M|S,k,\Psi) = \begin{pmatrix} P_t(\Psi_{S_1,1}) & \dots & P_t(\Psi_{S_k,1}) & \dots & P_t(\Psi_{S_N,1}) \\ P_t(\Psi_{S_1,2}) & \dots & P_t(\Psi_{S_k,2}) & \dots & P_t(\Psi_{S_N,2}) \\ \dots & \dots & \dots & \dots & \dots \\ P_t(\Psi_{S_1,N_{S_{max}}}) & \dots & P_t(\Psi_{S_k,N_{S_{max}}}) & \dots & P_t(\Psi_{S_N,N_{S_{max}}}) \end{pmatrix}, \quad (8)$$

where

$$P_t(\Psi_{S_k,j}) = \begin{cases} 0, if\ function\ \Psi_{S_k,j}\ is\ missing\ in\ S_k; \\ 1, if\ function\ \Psi_{S_k,j}\ is\ present\ in\ \ and\ is\ inactive; \\ 2, if\ function\ \Psi_{S_k,j}\ is\ available\ and\ is\ executed. \end{cases}$$

$\Psi_{S_k,j}$ – is a j-th function that is part of the set function Ψ_{S_k} of the component S_k; $k = 1,2, \dots, N$; N – the number of components in the system that are installed in computer stations in the network; k – number of system components in which the system center can be located; $S = S_1 \cup S_2 \cup \dots \cup S_k \cup \dots \cup S_N$; $j = 1,2, \dots, N_{S_{max}}$;

$N_{S_{max}}$ – is the largest number of functions that can be set in component S_k.

The bit map of active functions according to formula (8) defines a refined model of the architecture $M_{S,k,\Psi}$ of a partially centralized system S according to the functions available in it at the level of their execution at the current time. Functions can be executed pseudo-parallel, i.e., two or more functions can be executed simultaneously in one component. Thus, the matrix according to formula (8) will reflect the current state of functioning in all components. That is, maintaining up-to-date information about the state of the component and the component as a whole. Such information will be used by the decision-making center of the system.

In the matrix (formula (8)), the first *k* columns reflect the activity of the functions-subsets of the decision center of the system at the current time. The center of the system can be located directly in all *k* components and perform tasks involving all its parts from *k* components. At the same time, there may be duplication of tasks in different components, followed by analysis of the results, processing them to form and fix the resulting solution. Or, in *k* components, only certain functions may be involved in solving the system's tasks, i.e., not all the same functions in different components will be executed simultaneously. The order of their involvement in this case will be determined by the same system decision center. The system decision center may not be active simultaneously in all k components in the nodes in the network, but in a smaller number of components. In this case, the functionality for activating the decision-making center is contained in all *k* кcomponents of the system, respectively, and is activated by the system in the course of its operation. Such activation of the functionality responsible for system decision-making in the component makes it possible to move the system decision-making center between different components of the system from the defined *k* components in the course of system operation. This organized movement of the decision-making center with distribution among components, as well as with the possibility of moving entirely to one of the components, is part of the synthesis of the adaptability property in the system. The migration of the system's decision-making center will improve its resilience in the presence of cyberthreats.

Let's form a set of all possible different options for the location of the system's de-cision-making center, depending on the number of components in the system and the functions for ensuring the of the decision-making center executing. At this stage we will not take into account the security levels in the components in which the system decision center will be located. The security levels in these components will be taken into account when choosing options for placing the decision-making center in the process of system operation.

Then, let m be the number of active components in the system in which the system's decision-making center is currently functioning, $2 \le m \le k$, where k is the number of system components in which the system center can be located. The number of functions that will form the complete functional component of the decision-making center in the component in the system is equal to $n_{S_k,max}$, and $1 \le n_{S_k,max} < N_{S_{max}}$, where $N_{S_{max}}$ is the largest number of functions that can be installed in the component S_k. The matrix that will reflect the activity of the functions that ensure the functioning of the decision-making center at the current moment and their availability, according to formula (8), is given as follows:

$$P_{t,n_{S_k,max}}(M|S,k,\Psi,) = \begin{pmatrix} P_t(\Psi_{S_1,1}) & \cdots & P_t(\Psi_{S_k,1}) \\ P_t(\Psi_{S_1,2}) & \cdots & P_t(\Psi_{S_k,2}) \\ \cdots & \cdots & \cdots \\ P_t(\Psi_{S_1,n_{S_k,max}}) & \cdots & P_t(\Psi_{S_k,n_{S_k,max}}) \end{pmatrix}, \quad (9)$$

where

$$P(\Psi_{S_k,j}) = \begin{cases} 0, if\ function\ \Psi_{S_k,j}\ is\ missing\ in\ the\ component; \\ 1, if\ function\ \Psi_{S_k,j}\ is\ present\ in\ the\ component. \end{cases}$$

$\Psi_{S_k,j}$ – j-th function that is part of the set function Ψ_{S_k} of the component S_k; k – number of system components in which the system center can be located.

In order to form the set of options for the location of the system's decision-making center, we divide the typical ones into subsets. Then, let us define the subsets of options for the location of the decision-making center of the system as follows:

1. a subset Q_1 contains options with a choice of any *m* components from the available active components out of all possible *k* components;
2. a subset Q_2 contains *m* formed dynamic components from the functions of different *k* components without repetition in different components of the functions of subsets of a certain component; and *m* static components are formed by the functions of subsets of the remaining components; the functions will be selected from the available active components from the possible *k* components.

An example of the image of a dynamic component from functions of different k components is shown in Fig. 2.

Subset functions are shown in Fig. 2 by vertices and links between them by edges.

The dynamic components that will contain the decision-making center of the sys-tem are formed in the form of trees, and together they form a forest. The links between these trees as components are established in another component subsystem that is not directly connected to the decision-making center and does not participate in deci-sion-making. Dynamic components formed in this way allow for independent distrib-uted computing, which minimizes the impact on decision-making, including for mali-cious purposes.

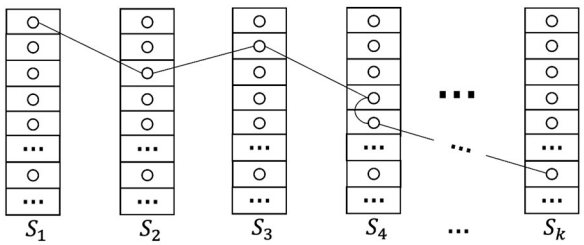

Figure 2. An example of a dynamic component with of functions of different k components.

Events in a partially centralized system, including the performance of functions and receipt of external influences for

processing, will not have the same degree of trust, and will not be performed under the same conditions in different nodes in the network in which the components are installed, and will not have the same capabilities when performed at the same computer station and according to a remote procedure call. Therefore, each time the functions in the components are performed to form the final result, the trust in the intermediate results should be taken into account, which can be expressed as a certain share or percentage compared to the result obtained un-der ideal conditions. In addition, computer stations in the network where components of a partially centralized system are installed and the processes performed in them may be influenced by different workloads, as well as by malicious software, which will lead to the system obtaining values of the system functions performed with different time frames and, under certain conditions, distorted results of values. In this regard, both computer stations, system components, and the values of the distributed computations performed need to be evaluated. All these estimates must be taken into account by the decision-making center of a partially centralized system. To evaluate them, we will introduce criteria that will form the level of trust in system components, component subsystems, functions, and the resulting evaluated values based on the results of function execution. The level of trust in the results of distributed computing obtained from different components of a partially centralized system may be different, and these values should affect the resulting computations.

Let's set the level of confidence R_S in the evaluation results, which will be used for distributed computing directly in the system, with values from the interval $[0; 1]$, i.e. $0 \le R_S \le 1$. Let us take the greater of the two values R_S as the one with the higher degree of confidence in the evaluation results.

Components of a partially centralized system are divided into those in which the system decision-making center may be located and those in which there are no means of its functioning. In addition, the components that may have a system decision-making center are divided into two subsets: the decision-making center functions in the component; the decision-making center does not function in the component. Therefore, the level of confidence in the results of evaluations from the system components will be determined taking into account this division. Let the level of confidence in the results of evaluations obtained from each component be given by values from the interval $[0; 1]$, i.e. $0 \le R_{S,S_k} \le 1$, where S_k – is the system component $k = 1,2, \dots, N$, N is the number of components in the system that are installed in computer stations in the network, k is the number of system components in which the system decision center can be located.

Calculations in a particular component can be performed taking into account the membership of functions in subsets in three ways:

1. evaluations that relate exclusively to the decision center of the system and are performed by the corresponding functions-subsets of the decision center;
2. 2)evaluations performed by functions that are not functions of the system's deci-sion center;
3. evaluations performed by functions of both the decision center of the system and the rest of the functions.

The first and second options provide for the execution of functions specifically for the decision center in the first option or for the remaining functions of the component. The third option involves the execution of functions that are not part of the decision cen-ter's functions. But, if necessary, later they may be used for malware detection. The result of such evaluations may be, for example, a value that requires either changing the sys-tem architecture or the location of the system's decision center or performing other steps built into the system. Separating such a variant into a separate one is inappro-priate, since it does not cause changes in the system as a whole in one cycle, as it can happen in the third variant. The results of the evaluations obtained by the second op-tion are sent to the decision-making center of the system.

To perform specific tasks, dynamic components can execute functions quantita-tively in different ways: one function; several functions ; all functions . At the same time, the functions that will form the decision-making center of the system will also be executed. Accordingly, in this case, such variants of execution will affect the level of confidence in the results of evaluations and should be taken into account when deter-mining it.

Let's set the weighting coefficients of the functions in the components in which the decision center of the system can be located as follows:

$$M_{\alpha_{1,S_k}} = \begin{pmatrix} \alpha_{1,S_1,1} & \cdots & \alpha_{1,S_k,1} \\ \vdots & \ddots & \vdots \\ \alpha_{1,S_1,n_{S_k,max}} & \cdots & \alpha_{1,S_k,n_{S_k,max}} \end{pmatrix}, \quad (10)$$

where $\alpha_{1,S_i,j}$ – the weighting coefficient of the functions in the components in which the decision-making center of the system can be located; $j = 1,2, \dots, n_{S_k,max}$; $n_{S_k,max}$- number of functions that will form a complete functional part of the decision-making center in the component in the system; $i = 1,2, \dots, k$; k – number of system components that can have a system decision-making center.

If the function is included in the dynamic component and is in the same static component, then the value of the weighting coefficient for it is set to one $\alpha_{1,S_i,j} = 1$. If the function is part of the dynamic component and is located in another static component, then the value of the weighting coefficient for it will be set in the range $(0; 1)$: $0 < \alpha_{1,S_i,j} < 1$. If the function is part of the dynamic component, but was not used to perform the task, then the value of the coefficient is set to zero $\alpha_{1,S_i,j} = 0$. Thus, when performing a certain decision-making task in the system, certain functions will be activated in the components to perform certain functions, and for these components, a matrix of weighting coefficients of the functions-subsets in the components will be formed, the values of which will be used to determine the level of confidence in the evaluation results.

Let's define the level of confidence r_{1,S_i} in the results of evaluations when performing a task in one of the dynamic components S_i of the system in which the decision-making center is located as follows:

$$r_{1,S_i} = \frac{\sum_{j=1}^{n_{S_k,max}} \alpha'_{1,S_i,j}}{n'_{1,S_i}}, \quad (11)$$

where $\alpha'_{1,S_i,j}$ - is the weighting coefficient of a function of the dynamic component; n'_{1,S_i} - the number of functions that were performed during the evaluations; $j = 1,2, \dots, n_{S_k,max}$; $n_{S_k,max}$ - number of functions that will form a complete functional part

of the decision-making center in the component in the system; $i = 1,2,\dots,k$; k – the number of system components that can have a system decision-making center; $i \le k$.

In the values of these coefficients, we will also take into account the order of function execution of in dynamic components, the time spent on sending execution results, the security level of the node, the execution of functions in components with inactive subsystems of the system's decision center, and the number of functions that participated in the task. Then, the weights of the functions in the dynamic components of S_i are defined as functions with five arguments as follows:

$$\alpha'_{1,S_i} = f_{\alpha'_{1,S_i}}\left(\alpha'_{1,S_i,1}, \alpha'_{1,S_i,2}, \alpha'_{1,S_i,3}, \alpha'_{1,S_i,4}, \alpha'_{1,S_i,5}\right), \quad (12)$$

where $\alpha'_{1,S_i,1}$ - a value that takes into account the order of execution of functions in dynamic components; $\alpha'_{1,S_i,2}$ - is a value that takes into account the relative time spent on sending execution results; $\alpha'_{1,S_i,3}$ - a value that takes into account the security level of the node in the network; $\alpha'_{1,S_i,4}$ - is a value that takes into account the execution of functions in components with inactive subsystems of the system's decision center; $\alpha'_{1,S_i,5}$ - a value that takes into account the number of functions that participated in the task; S_i - i – component of the system; $i = 1,2,\dots,k$; k – the number of system components that can have a system decision-making center; $i \le k$.

The value of $\alpha'_{1,S_i,1}$ takes into account the order of execution of functions in dynamic components and depends on the order of execution of a particular j function in the general order of execution of functions. Let's assume that all the functions that form the decision-making center of the system were involved in the task and that they were all located in one node in the network. The number of functions is determined by the value of $n_{S_k,max}$, then the vector $v_{\alpha'_{1,S_i}}$ sets the order of execution of the functions in the S_i component for a certain period of time when a certain task is fully completed. Let us denote a function $f_{nom}(j)$, whose value is the number of j – the function in the list of functions. As a result of this definition, the vector $v_{\alpha'_{1,S_i}} = \left(f_{nom}(1), f_{nom}(2), \dots, f_{nom}\left(n_{S_k,max}\right)\right)$. The order numbers of the functions are positive integers from the interval $\left[1, n_{S_k,max}\right]$, and their sum will be the value of the arithmetic progression. For the values of $\alpha'_{1,S_i,1}$ let's introduce a range within which the lower bound will be adjusted depending on the significance level parameter $\alpha_1^{r,1}$ as follows: $\left[1-\alpha_1^{r,1}; 1\right]$. The level of significance is a fraction of one, which reflects deviations from the level of confidence in the evaluation result due to certain events, architectural features of the component, etc. The value of $\alpha'_{1,S_i,1}$ is determined by taking into account the coordinates of the vector $v_{\alpha'_{1,S_i}}$, which are normalized to the interval $\left[1-\alpha_1^{r,1}; 1\right]$, as follows:

determine the largest value from the coordinates of the vector $v_{\alpha'_{1,S_i}}$:
$f_{nom,max} = max\left(f_{nom}(1), f_{nom}(2), \dots, f_{nom}\left(n_{S_k,max}\right)\right)$;
is the smallest value of the coordinates of the vector $v_{\alpha'_{1,S_i}}$:
$f_{nom,min} = 1$;
let us determine the step for placing normalized values from the interval $\left[f_{nom,min}; f_{nom,max}\right]$ to the interval $\left[1-\alpha_1^{r,1}; 1\right]$, dividing the interval into equal segments and their number should correspond to the number of involved functions:
$\frac{1-(1-\alpha_1^{r,1})}{n_{S_k,max}-1} = \frac{\alpha_1^{r,1}}{n_{S_k,max}-1}$;
let us evaluate the value of $\alpha'_{1,S_i,j,1}$ for the j-th function as follows:

$$\alpha'_{1,S_i,j,1} = 1 - \frac{\left(n_{S_k,max} - f_{nom}(j)\right)}{n_{S_k,max}-1} \cdot \alpha_1^{r,1}, \quad (13)$$

where $j = 1,2,\dots,n_{S_k,max}$; $n_{S_k,max}$ - number of functions that will form a complete functional part of the decision-making center in the component in the system; $i = 1,2,\dots,k$; k – the number of system components that can have a system decision-making center; $i \le k$; $f_{nom}(j)$ - is a function whose value is the number of the j -th function in the list of functions;
let us determine the value of $\alpha'_{1,S_i,1}$ according to step 4 (formula (13)):

$$\alpha'_{1,S_i,1} = \frac{\sum_{j=1}^{n_{S_k,max}} \alpha'_{1,S_i,j,i}}{n_{S_k,max}}, \quad (14)$$

The values of $\alpha'_{1,S_i,j,1}$ evaluated by formula (13) will be larger for those functions that are executed earlier than the others.

Let us clarify formula (14) for the general case when the number of functions may be less than $n_{S_k,max}$, some of the functions can be executed multiple times, some of the functions can be and are executed in other components in which the decision-making center of the system is active. If the number of functions that will be executed to solve the problem is less than $n_{S_k,max}$, then we will introduce the variable $n_{S_k,max,f,1}$, which will set the number of functions involved, and the variable $n_{S_k,max,f,2}$, which will set the number of functions involved, and will take into account the multiple use of functions, if such an execution occurs. Let's define the vector $v_{\alpha'_{1,S_i},f,1} = \left(f_{nom,f}(1), f_{nom,f}(2), \dots, f_{nom,f}\left(n_{S_k,max,f,2}\right)\right)$ so that its coordinates are able to indicate the number of the function in the dynamic component that was executed. The numbers of the order of function execution are positive integers from the interval $\left[1, n_{S_k,max,f,2}\right]$. In addition, the coordinate numbers of the vector $v_{\alpha'_{1,S_i},f,1}$ determine the sequence of execution of the functions. We form a vector of coordinate numbers (the value of the function $f_{nom,f,2}$ is the coordinate number) of the vector $v_{\alpha'_{1,S_i},f,1}$ as follows:

$v_{\alpha'_{1,S_i},f,2} = \left(f_{nom,f,2}(1), f_{nom,f,2}(2), \dots, f_{nom,f,2}\left(n_{S_k,max,f,2}\right)\right)$.

According to the values in the generated vectors $v_{\alpha'_{1,S_i}}$, $v_{\alpha'_{1,S_i},f,1}$, $v_{\alpha'_{1,S_i},f,2}$ we set separate vectors for each function, the coordinates of which will be the numbers obtained as a result of the task and set the sequence of their execution, including multiple calls. For the j-th function, the vector will have a number of coordinates equal to the number of its calls $K_{f,j}$, and we define it as follows: $v_{\alpha'_{1,S_i},j} = \left(f_{nom}(j,1), f_{nom}(j,2), \dots, f_{nom}\left(j, K_{f,j}\right)\right)$. The order of the coordinates of this vector will correspond to the sequence of calls. The number of executed functions when solving a certain task is always finite and the number of such calls to functions is equal to $n_{S_k,max,f,2}$. For functions-subsets that will be executed multiple times when performing the task, the minimum value of $\alpha'_{1,S_i,j,1}$ will be selected from the evaluated ones. The values of $\alpha'_{1,S_i,j,1}$ for the functions that were not involved in the task are set to zero. Thus, the evaluation of the value of $\alpha'_{1,S_i,1}$ taking into account the added requirements, can be done as follows:

$$\alpha'_{1,S_i,j,1} = \min_{q=1,2,\dots,K_{f,j}} \left(1 - \frac{n_{S_k,max,f,2} - f_{nom}(j,q)}{n_{S_k,max,f,2} - 1} \cdot \alpha_1^{r,1}\right), \quad (15)$$

where, after evaluation the values for each function, the results are substituted into formula (2.22.2).

Let's simplify formula (15) to take into account the use of the minimum value of $f_{nom}(j,q)$ (at $q = 1,2,\dots,K_{f,j}$). Since the first value is the smallest, we will evaluate the value of $\alpha'_{1,S_i,j,1}$ for each function as follows:

$$\alpha'_{1,S_i,j,1} = 1 - \frac{n_{S_k,max,f,2} - f_{nom}(j,1)}{n_{S_k,max,f,2} - 1} \cdot \alpha_1^{r,1}, \quad (16)$$

where $f_{nom}(j,1)$ – is the first smallest value of the coordinate of the vector $v_{\alpha'_{1,S_i},j}$; $n_{S_k,max,f,2}$ - the number of functions performed when solving a particular task.

Then, similarly, after evaluation the values for each function using formula (16), we substitute the results into formula (14).

If some of the functions can be and are performed in the rest of the system components in which the system decision center is active, then the level of confidence in the results of evaluations compared to evaluations performed in one component will be different. The number of such functions will affect the overall result and the individual results for each of them, and some of them can be called for execution multiple times, so for each of them, when determining the value of $\alpha'_{1,S_i,j,1}$ we will take into account their number, the number of multiple calls to certain functions, and the level of significance as follows:

$$\alpha'_{1,S_i,j,1} = 1 - \left(\frac{n_{S_k,max,f,2} - f_{nom}(j,1)}{n_{S_k,max,f,2} - 1} + \frac{K_{f,j}}{n_{S_k,max,f,2}}\right) \cdot \alpha_1^{r,1}, \quad (17)$$

where $f_{nom}(j,1)$ – is the first smallest value of the coordinate of the vector $v_{\alpha'_{1,S_i},j}$; $n_{S_k,max,f,2}$ - the number of subsets involved in performing functions when solving a particular task; $K_{f,j}$ – is the number of calls to the j-th function.

Similarly, after evaluation the values for each function using formula (17), the results are substituted into formula (14).

Thus, according to formulas (17) and (14), the gap $\left[1 - \alpha_1^{r,1}; 1\right]$ for functions can be expanded in the lower bound. For functions that are launched for execution in parallel, the values of the launch sequence will be the same, and the next function launched for execution after them will have the launch number increased by one.

Let's determine the value of the second coordinate of the vector in formula (12) $\alpha'_{1,S_i,2}$, which takes into account the relative time spent on sending the results of the execution of functions. Let's assume that the functions were used only once during the task execution. Let us define the function $f_{nom,t}(j)$, the value of which is the time spent on sending the results of the evaluation of the $j-$th function to the dynamic component. As a result of this definition, we form a vector $v_{\alpha'_{1,S_i},t} = \left(f_{nom,t}(1), f_{nom,t}(2), \dots, f_{nom,t}\left(n_{S_k,max}\right)\right)$. The coordinate numbers of the vector $v_{\alpha'_{1,S_i},t}$ are numbers from the interval $\left[1, n_{S_k,max}\right]$. For the values of $\alpha'_{1,S_i,j,2}$ of the j -th function, let's introduce a range $\left[1 - \alpha_1^{r,2}; 1\right]$, in which the lower bound will be adjusted depending on the significance level parameter $\alpha_1^{r,2}$ by the following number $1 - \alpha_1^{r,2}$. The level of significance is a fraction of one, which reflects the deviation from the level of confidence in the evaluation result. The value of $\alpha'_{1,S_i,j,2}$ $(j = 1,2,\dots,n_{S_k,max})$ is determined taking into account the coordinates of the vector $v_{\alpha'_{1,S_i},t}$, which are normalized to the interval $\left[1 - \alpha_1^{r,2}; 1\right]$, as follows:

to determine the largest value from the coordinates of the vector $v_{\alpha'_{1,S_i},t}$:

$f_{nom,max,t} = max\left(f_{nom,t}(1), f_{nom,t}(2), \dots, f_{nom,t}\left(n_{S_k,max}\right)\right)$;

to determine the smallest value from the coordinates of the vector $v_{\alpha'_{1,S_i},t}$:

$f_{nom,min,t} = min\left(f_{nom,t}(1), f_{nom,t}(2), \dots, f_{nom,t}\left(n_{S_k,max}\right)\right)$;

to evaluate the value of $\alpha'_{1,S_i,j,2}$ for the j-th function at $j = 1,2,\dots,n_{S_k,max}$ as follows:

$$\alpha'_{1,S_i,j,2} = 1 - \alpha_1^{r,2} + \frac{\left(f_{nom,max,t} - f_{nom,t}(j)\right)}{f_{nom,max,t}} \cdot \alpha_1^{r,2} = 1 - \frac{f_{nom,t}(j)}{f_{nom,max,t}} \cdot \alpha_1^{r,2}, \quad (18)$$

where $n_{S_k,max}$ - number of functions that will form a complete functional part of the decision-making center in the component in the system; $i = 1,2,\dots,k$; k – the number of system components that can have a system decision-making center; $i \le k$;

4)to determine the value of $\alpha'_{1,S_i,2}$ according to step 3 (formula (18)):

$$\alpha'_{1,S_i,2} = \frac{\sum_{j=1}^{n_{S_k,max}} \alpha'_{1,S_i,j,i}}{n_{S_k,max}}, \quad (19)$$

Formula (19) makes it possible to perform evaluations for the case when some of the functions are placed directly in the component in the node in the network, then according to it, the evaluated value of $\alpha'_{1,S_i,j,2}$ for such functions is equal to one, since no time is spent on sending the results of evaluations.

Let's clarify formula (19) for cases when the functions are executed multiple times and when the number of functions that will be executed to solve the task may be less than $n_{S_k,max}$. If the number of functions that will be executed to solve the task is less than $n_{S_k,max}$, then we introduce the variable $n_{S_k,max,f,1,t}$, which will set the number of functions involved, and the variable $n_{S_k,max,f,2,t}$, which will set the number of functions involved, and will take into account the multiple involvement of functions, if such an execution occurs. Let us define the vector

$v_{\alpha'_{1,S_i},f,1,t} = \left(f_{nom,f,t}(1), f_{nom,f,t}(2), \dots, f_{nom,f,t}\left(n_{S_k,max,f,2,t}\right)\right)$ so that its coordinates will indicate the number of the function in the dynamic component that was executed. The order numbers of the functions are positive integers from the interval $\left[1, n_{S_k,max,f,2,t}\right]$. In addition, the coordinate numbers of the vector $v_{\alpha'_{1,S_i},f,1,t}$ determine the sequence of execution of the functions. Let us form the vector of coordinate numbers of vector $v_{\alpha'_{1,S_i},f,1,t}$ as follows:

$v_{\alpha'_{1,S_i,f,2,t},f,2,t} = \left(f_{nom,f,2,t}(1), f_{nom,f,2,t}(2), \dots, f_{nom,f,2,t}\left(n_{S_k,max,f,2,t}\right)\right)$.

According to the values in the generated vectors $v_{\alpha'_{1,S_i},t}$, $v_{\alpha'_{1,S_i},f,1,t}$, $v_{\alpha'_{1,S_i},f,2,t}$ we set separate vectors for each function, the coordinates of which will be the numbers obtained as a result of the task and setting the sequence of their execution, including multiple calls. For the j-th function, the vector will have a

number of coordinates equal to the number of its calls $K_{f,j,t}$, and we define it as follows: $v_{\alpha'_{1,S_i},j,t} = \left(f_{nom,t}(j,1), f_{nom,t}(j,2), \dots, f_{nom,t}\left(j,K_{f,j,t}\right)\right)$. The order of coordinates of this vector will correspond to the sequence of calls. The number of executed functions when solving a particular task is always finite and the number of such calls to functions is equal to $n_{S_k,max,f,2,t}$. The values of the coordinates of the vector $v_{\alpha'_{1,S_i},j,t}$ at $j = 1,2,\dots,n_{S_k,max}$ will be the values of the time spent on sending the evaluation results. For functions that will be executed repeatedly when performing the task, the total value of the time spent on execution will be selected. The values of $\alpha'_{1,S_i,j,2}$ for the functions that were not involved in the task are set to zero. Thus, the value of $\alpha'_{1,S_i,j,2}$ is evaluated as follows:

$$\alpha'_{1,S_i,j,2} = 1 - \frac{\sum_{q=1}^{K_{f,j,t}} f_{nom,t}(j,q)}{\sum_{j=1}^{n_{S_k,max}} \sum_{q=1}^{K_{f,j,t}} f_{nom,t}(j,q)} \cdot \alpha_1^{r,2}, \qquad (20)$$

where $f_{nom,t}(j,q)$ –is the value of the coordinate of the vector $v_{\alpha'_{1,S_i},j,t}$; $n_{S_k,max}$ - the number of functions in the component; $K_{f,j,t}$ – is the number of calls to the j-th function.

According to formula (19), substituting the values of $\alpha'_{1,S_i,2}$ obtained by formula (20), we evaluate the values for the components of S_i ($i = 1,2,\dots,k$; k – is the number of system components in which the system decision-making center can be located; $i \le k$).

Thus, according to formula (18), the value of $\alpha'_{1,S_i,j,2}$ is evaluated taking into account the ratio of the time spent on sending the evaluation result for one function to the maximum time, which is defined as the maximum value of the time spent for individual functions. And according to formula (20), the value of $\alpha'_{1,S_i,j,2}$ is determined by the ratio of the total time spent by one function, including multiple calls, to the total time spent by the functions to solve the task. Therefore, the values of $\alpha'_{1,S_i,j,2}$, obtained by formula (20) will be larger than those obtained by formula (18) for a single execution of the functions.

Let's define $\alpha'_{1,S_i,3}$ as a value that takes into account the security level of a node in the network and contains data on the security status of a node in the network and is determined directly by a partially centralized system. This value will also be updated during the activity of the system component. The value of $\alpha'_{1,S_i,3}$ itself will be determined from the interval $\left[1-\alpha_1^{r,3}; 1\right]$, where $\alpha_1^{r,3}$ is the level of significance. Confidence in the results of evaluations from a particular component of the system will depend on the value of $\alpha'_{1,S_i,3}$. The level of significance is a fraction of one. The significance level $\alpha_1^{r,3}$ will apply to the evaluations that will be carried out at the level of the decision-making center. Its value will be greater than the value of the significance level for evaluations that do not belong to the level of the decision-making center, i.e. for the rest of the evaluations. This is because the decisions of the decision center determine the sustainability of the entire partially centralized system and its further steps. The value of $\alpha'_{1,S_i,3}$ will be evaluated taking into account the criteria for a comprehensive assessment of the security of a node in the network where S_i components of the system will be located. To do this, let's consider separately the following components that will affect the value of $\alpha'_{1,S_i,3}$: processes that occur at the level of the network and network services; processes that occur in the computer station; internal processes in a partially centralized system; the security level value that will be issued directly by the partially centralized system.

Network services, placement of stations in a computer network, configuration of a computer network and other characteristic components related to the functioning of nodes in networks will significantly affect security, and therefore the value of $\alpha'_{1,S_i,3}$. The component that will form the value of $\alpha'_{1,S_i,3}$, and set separate intermediate values that describe the processes that occur at the level of the network and network services is multicomponent, and each component has different characteristics. Based on the planned location of the proposed partially centralized system and the tasks it will have to perform, the security level will be formed, starting with the location in computer networks and their configuration features.

The location of a partially centralized system is the corporate computer network of an enterprise, organization or institution. A certain group of users will have access to network resources remotely. For example, at home. Therefore, the peculiarities of different types of architectural configurations of computer networks that may be present in the enterprise must be taken into account when determining the values of the security level, and the system must have information about them and the value of the security level depending on the equipment and types of configurations.

To determine the level of security of a partially centralized system, let's take into account the user and network resource management models used directly in the organization of the enterprise's computer networks. A distributed management model (workgroup management model) compared to a centralized management model (domain management model) with a large number of computer stations in an enterprise network will have a negative impact on network security. This is due to the fact that a workgroup is a logical group of computers for providing access to resources and is used in peer-to-peer networks, where each computer station stores a local security database, including information about user accounts and resource protection. For a small number of computers, this distributed management model has an advantage over the centralized management model. As a rule, the number of computers in such a model should be no more than 8-10 to ensure efficient use of network resources and, accordingly, safe operation. If the number is higher, the network built using the centralized management model has an advantage. In enterprises, the computer network in which System S, is to be installed can be built using different models, including inefficient ones (for example, according to a decentralized management model with more than 20 computers), but System S must have this information to determine its next steps, which will be based on an assessment of the security status of the network as a whole and its specific nodes. Let's introduce a parameter to define a control model in the system and denote it by $\alpha''_{1,S_i,3,1}$. The value of $\alpha''_{1,S_i,3,1} = 1$, if the network is built according to the distributed management model with the number of computers not exceeding 8 and $\alpha''_{1,S_i,3,1} = 2$ – not less than 9, and $\alpha''_{1,S_i,3,1} = 3$, if the network is built according to the centralized management model. Let's introduce a local significance level $\alpha_1^{r,3,1}$ for the significance level $\alpha_1^{r,3}$, which will reflect the deviation from the level of confidence in the evaluation result. Let us take a fraction

of one as the local level of significance $\alpha_1^{r,3,1}$ then the values of confidence in the results of evaluations for all nodes in the network will be the same and will be in the range $[1-\alpha_1^{r,3,1};1]$. The value of $\alpha''_{1,S_i,3,1}$ ($i=1,2,\dots,k$; k – the number of system components that can have a system decision-making center; $i \le k$) is defined as follows:

$$\alpha'_{1,S_i,3,1}=\begin{cases}1, if\, \alpha''_{1,S_i,3,1}=1;\\ 1-\frac{n_{1,S_i,3,1,k}-n_{1,S_i,3,1,p}}{n_k}\cdot\alpha_1^{r,3,1}, if\, \alpha''_{1,S_i,3,1}=2;\\ 1-\frac{n_{1,S_i,3,1,s}-1}{n_{1,S_i,3,1,k}}\cdot\alpha_1^{r,3,1}, if\, \alpha''_{1,S_i,3,1}=3,\end{cases} \quad (21)$$

where $n_{1,S_i,3,1,k}$ – the number of computer stations in the network; $n_{1,S_i,3,1,p}$ – the number of powered off computer stations in the network; $n_{1,S_i,3,1,s}$ – number of segments in the network.

Similarly, let's define the rest of the characteristic indicators to evaluate the value of the security level.

When authenticating users, there may be failed attempts and its accumulation can be the reason for denial of access. But it can also be accepted as an attacker's attempts to log in using the login of an authorized user or frequent false attempts by an unqualified user. In the first case, the security level cannot be maximized because the attacker has gained physical access to the authentication system, and in the second case, the erroneous actions of a user during authentication indicate that he or she may perform the rest of the work in the information system, including those related to security, in an unskilled manner. Let's introduce a component $\alpha'_{1,S_i,3,2}$, that will form the value of $\alpha'_{1,S_i,3}$, and its evaluated value will be a characteristic authentication indicator. Let us define the value of $\alpha'_{1,S_i,3,2}$ for each i-th computer station so that it belongs to the interval $[1-\alpha_1^{r,3,2};1]$, where the local level of significance $\alpha_1^{r,3,2}$ is a fraction of one and reflects the deviation from the level of confidence in the result of the evaluation. Its value is evaluated by the formula:

$$\alpha'_{1,S_i,3,2}=1-\frac{n_{1,S_i,3,2,k}-n_{1,S_i,3,2,p}}{n_{1,S_i,3,2,k}}\cdot\alpha_1^{r,3,2}, \quad (22)$$

where $n_{1,S_i,3,2,k}$ – is the number of attempts of users to authenticate in the information system at the i-th computer station; $n_{1,S_i,3,2,p}$ – is the number of successful attempts of users to authenticate in the information system at the i-th computer station.

False recognition of an unauthorized user by a computer station, which results in accidental login and access to it, is characterized by the fact that the user registry records the login of an unauthorized user and the granting of access. Let's introduce a component $\alpha'_{1,S_i,3,3}$, that will form the value of $\alpha'_{1,S_i,3}$, and its evaluated value will be a characteristic indicator of the false recognition of a third-party user. The value of $\alpha'_{1,S_i,3,3}$ will be determined for each i-th computer station so that it belongs to the interval $[1-\alpha_1^{r,3,3};1]$, where the local level of significance $\alpha_1^{r,3,3}$ is a fraction of one and reflects the deviation from the level of confidence in the result of the evaluation. Its value is evaluated by the formula:

$$\alpha'_{1,S_i,3,3}=1-\frac{n_{1,S_i,3,3,p}}{n_{1,S_i,3,3,k}}\cdot\alpha_1^{r,3,3}, \quad (23)$$

where $n_{1,S_i,3,3,k}$ – is the number of attempts by unauthorized users to log in to the i-th computer station; $n_{1,S_i,3,3,p}$ – is the number of successful attempts by third-party users to log in to the i-th computer station.

The values of $n_{1,S_i,3,3,k}$ and $n_{1,S_i,3,3,p}$ can be obtained when the components of the system S are installed in all nodes in the network and during a certain time of operation of the system S.

Let's consider the false denial of an authorized user when he tries to access the j-th computer station. Let's introduce a component $\alpha'_{1,S_i,3,4}$, that will form the value of $\alpha'_{1,S_i,3}$, and its value will be a characteristic indicator of the authorized user's false denial. The value of $\alpha'_{1,S_i,3,4}$ will be determined for each i-th computer station so that it belongs to the interval $[1-\alpha_1^{r,3,4};1]$, where the local level of significance $\alpha_1^{r,3,4}$ is a fraction of one and reflects the deviation from the level of confidence in the result of the evaluation. The value is evaluated as follows:

$$\alpha'_{1,S_i,3,4}=1-\frac{n_{1,S_i,3,4,p}}{n_{1,S_i,3,4,k}}\cdot\alpha_1^{r,3,4}, \quad (24)$$

where $n_{1,S_i,3,3,k}$ – is the number of attempts by authorized users to log in to the i-th computer station; $n_{1,S_i,3,3,p}$ – is the number of unsuccessful attempts by authorized users to log in to the i-th computer station.

Authorized users should be provided with separate access to computer station resources. However, there may not be such a distinction. In addition, the number of authorized users who have access to and work with one computer station may be different. The company may have different schemes for differentiating user authentication access to resources. Also, access control to resources should take into account the possibility of certain employees of the enterprise performing work from home computers at a certain time. The specifics of working with access control and levels of authorized user separation should be taken into account when evaluation the security level. Let's introduce a component $\alpha'_{1,S_i,3,5}$, that will form the value of $\alpha'_{1,S_i,3}$, and its value will be a characteristic indicator of the organization of differentiation of access of authorized users to the resources of each i-th computer station in the enterprise network. If access is not differentiated, then the value is $\alpha''_{1,S_i,3,5}=1$, otherwise $\alpha''_{1,S_i,3,5}=2$. We define the value $\alpha'_{1,S_i,3,5}$ for each i-th computer station so that it belongs to the interval $[1-\alpha_1^{r,3,5};1]$, where the local significance level $\alpha_1^{r,3,5}$ is a fraction of one and reflects the deviation from the level of confidence in the evaluation result. The value is evaluated as follows:

$$\alpha'_{1,S_i,3,5}=\begin{cases}1-\alpha_1^{r,3,5}, if\, \alpha''_{1,S_i,3,5}=1\\ 1-\frac{n_{1,S_i,3,5,k}}{n_{1,S_i,3,5,p}}\cdot\alpha_1^{r,3,5}, if\, \alpha''_{1,S_i,3,5}=2\end{cases} \quad (25)$$

where $n_{1,S_i,3,5,k}$ – is the number of authorized users who are allowed and who have logged in to the i-th computer station with different access levels during a certain period of time; $n_{1,S_i,3,5,p}$ – is the number of authorized users who are allowed to log in to the i-th computer station with different access levels.

Some of the authorized users of the enterprise may be allowed to connect and log in from home computers. In this case, the value of the local significance level will be the same for all nodes in the network. Let's introduce the component $\alpha'_{1,S_i,3,6}$,

which will form the value of $\alpha'_{1,S_i,3}$, and its value will be a characteristic indicator of authorized users' access to resources on the enterprise network from home computers. If such access is not allowed due to a ban, then the value is $\alpha''_{1,S_i,3,6} = 1$, otherwise $\alpha''_{1,S_i,3,6} = 2$. The value of $\alpha'_{1,S_i,3,6}$ is defined for all computer stations so that it belongs to the interval $[1 - \alpha_1^{r,3,6}; 1]$, where the local level of significance $\alpha_1^{r,3,6}$ is a fraction of one and reflects the deviation from the level of confidence in the evaluation result. The evaluation of the value $\alpha'_{1,S_i,3,6}$ is as follows:

$$\alpha'_{1,S_i,3,6} = \begin{cases} 1, if\, \alpha''_{1,S_i,3,6} = 1 \\ 1 - \frac{n_{1,S_i,3,6,k}}{n_{1,S_i,3,6,p}} \cdot \alpha_1^{r,3,5}, if\, \alpha''_{1,S_i,3,6} = 2, \end{cases} \quad (26)$$

where $n_{1,S_i,3,6,k}$ – the number of authorized users who are allowed and have logged in to the company's system from outside from home computers with different access levels over a certain period of time; $n_{1,S_i,3,6,p}$ – the number of authorized users who are allowed to log in to the company's system from outside from home computers with different access levels.

The complexity of corporate networks affects the value of security levels in general and in their nodes in particular. Taking into account the diversity and multidirectionality of architectural features, we will define them as elements of the set $M_{A,KM}$. Combinations of them will reflect the formed corporate networks with architectural features that will be determined by the list of elements of a subset of the set $M_{A,KM}$. Thus, we set the following values for the elements of $M_{A,KM}$: $m_{A,KM,1}$ - the corporate network has a single connection to the Internet; $m_{A,KM,2}$ - the corporate network has several connections to the Internet; $m_{A,KM,3}$ - the corporate network has a long-standing unchanged architecture; $m_{A,KM,4}$ - the corporate network is located in the same premises of the enterprise; $m_{A,KM,5}$ - the corporate network is located in several premises of the enterprise directly on its territory and the network segments are connected by wired communication; $m_{A,KM,6}$ - the corporate network is located in several premises of the enterprise directly on its territory and the network segments are connected by wireless communication; $m_{A,KM,7}$ – the connection between the network segments is wired; $m_{A,KM,8}$ - communication between network segments is wireless; $m_{A,KM,9}$ – communication between some network nodes is wireless; $m_{A,KM,10}$ - the connection between all network nodes is wired; $m_{A,KM,11}$ - the corporate network has a frequently changing architecture, settings, users, etc; $m_{A,KM,12}$ - the corporate network consists of geographically separated components, and these components are connected by external networks located outside the company's territory and serviced by third-party network service providers; $m_{A,KM,13}$ – corporate network servers are located on the territory of the enterprise; $m_{A,KM,14}$ – all servers of corporate networks are located on the territory of the enterprise; $m_{A,KM,15}$ – all subsystems and elements of the corporate network are maintained and controlled by the company's administrators; $m_{A,KM,16}$ – not all subsystems and elements of the corporate network are maintained and controlled by enterprise administrators; $m_{A,KM,17}$ – a certain user uses resources that are permanently located in one place on the corporate network; $m_{A,KM,18}$ – a certain user uses resources located in different locations of the corporate network, including geographically remote locations, and with different platforms; $m_{A,KM,19}$ – access to the network and its resources should be organized around the clock; $m_{A,KM,20}$ – ensuring the requirement for the maximum downtime, which must be minimized and within the specified limits, for example, one minute. The number of elements of the set $M_{A,KM}$ can be increased by introducing and highlighting the rest of the architectural features of corporate networks. Let $n_{M_{A,KM}}$ be the possible maximum number of architectural features of corporate networks. According to the formed set $M_{A,KM}$, let us define, for example, its subsets $M_{A,KM,1} = \{m_{A,KM,2}, m_{A,KM,11}, m_{A,KM,19}\}$ and $M_{A,KM,2} = \{m_{A,KM,2}, m_{A,KM,13}, m_{A,KM,14}, m_{A,KM,20}\}$. The values of security levels in them will be different. Let the value of $\alpha''_{1,S_i,3,7} = 0$ be for the case of an element (for example, elements $m_{A,KM,1}$, $m_{A,KM,3}$, $m_{A,KM,4}$, $m_{A,KM,5}$, $m_{A,KM,7}$, $m_{A,KM,10}$, $m_{A,KM,13}$, $m_{A,KM,15}$, $m_{A,KM,17}$), that does not affect the reduction of the security level value compared to the second alternative element, which implies complications in the architecture of corporate networks. Then, the value of $\alpha''_{1,S_i,3,7} = 1$ will be accepted for the alternative case. Subsets that reflect realistic corporate network architectures may not be formed from all elements, i.e., not all elements can be combined when forming subsets, and therefore certain subsets may not exist. Let's introduce a function $f_{M_{A,KM}}$ whose value is given as follows:

$$f_{M_{A,KM}}(m_{A,KM,q}) = \begin{cases} 0, if\, \alpha''_{1,S_i,3,7} = 0 \\ 1, if\, \alpha''_{1,S_i,3,7} = 1 \end{cases} \quad (27)$$

where $q = 1,2,\dots,n_{M_{A,KM}}$; $n_{M_{A,KM}}$ - maximum number of architectural features of corporate networks.

The value of the local significance level for the characteristic indicator of the complexity of the corporate network architecture will be the same for all nodes in the network. Let's introduce the component $\alpha'_{1,S_i,3,7}$, which will form the value of $\alpha'_{1,S_i,3}$, and its value will be a characteristic indicator of the complexity of the corporate network architecture. Let us define the value $\alpha'_{1,S_i,3,7}$ for all computer stations so that it belongs to the interval $[1 - \alpha_1^{r,3,7}; 1]$, where the local level of significance $\alpha_1^{r,3,7}$ is a fraction of one and reflects the deviation from the level of confidence in the evaluation result. Let's evaluate the value $\alpha'_{1,S_i,3,7}$ as follows:

$$\alpha'_{1,S_i,3,7} = 1 - \frac{\sum_{w=1}^{n_{M_{A,KM,p}}} f_{M_{A,KL,p}}(m_{A,KM,w})}{\sum_{q=1}^{n_{M_{A,KM}}} f_{M_{A,KL}}(m_{A,KM,q})} \cdot \alpha_1^{r,3,7} \quad (28)$$

where $m_{A,KM,q}$ – set elements $M_{A,KL}$; $m_{A,KM,w}$ - elements of a subset $M_{A,KM,p}$; $n_{M_{A,KM,p}}$ – is the number of elements of the subset $M_{A,KM,p}$; $n_{M_{A,KM}}$ - is the number of elements of the set $M_{A,KL}$.

The availability of information protection and security tools is an integral part of corporate networks. An important element of the security service in a corporate network is firewalling, both for the computer station and the corporate network as a whole. There are several practical implementations of firewalls. Let's introduce a set $M_{E,KM}$, whose elements correspond to certain typical firewall implementations: $m_{E,KM,1}$ – with packet filtering; $m_{E,KM,2}$ – with a dual gateway; $m_{E,KM,3}$ – with an isolated host; $m_{E,KM,4}$ – with an isolated subnet. Let $n_{M_{E,KM}}$ – be the number of elements of the set $M_{E,KM}$. Subsets of $M_{E,KM}$ may contain one or more of the elements. This will determine the

complexity and security of the corporate network. Also, a corporate network may have several firewall implementations, including identical ones, so we will define such cases as a combination of subsets. Different implementation options may exist, for example, in geographically distributed parts of the corporate network. In this case, some of the elements of the set $M_{E,KM}$ may be repeated in different subsets. The evaluation of the security level value will be carried out taking into account all available subsets that characterize the implementation of network screens in the corporate network. The local level of significance for the characteristic indicator of the implementation of firewalls in corporate networks will be the same for all nodes in the network part with the same implementation. Let's introduce the component $\alpha'_{1,S_i,3,8}$, which will form the value $\alpha'_{1,S_i,3}$, and its value will be a characteristic indicator of the implementation of firewalls in a corporate network. We define the value of $\alpha'_{1,S_i,3,8}$ for all computer stations so that it belongs to the interval $\left[1-\alpha_1^{r,3,8};1\right]$, where the local level of significance $\alpha_1^{r,3,8}$ is a fraction of one and reflects the deviation from the level of confidence in the evaluation result. Let's introduce a function $f_{M_{E,KM}}$ whose value is given as follows:

$$f_{M_{E,KM}}\left(m_{E,KM,q}\right)=\frac{1}{q}, \tag{29}$$

where $q=1,2,\dots,n_{M_{E,KM}}$; $n_{M_{E,KM}}$ - is the number of elements of the set $M_{E,KM}$.
Then, let's evaluate the value of $\alpha'_{1,S_i,3,8}$ as follows:

$$\alpha'_{1,S_i,3,8}=1-\frac{n_{1,S_i,3,8,k}}{n_{1,S_i,3,8,p}}\cdot\prod_{w=1}^{n_{M_{E,KM,p}}}f_{M_{E,KM}}\left(m_{E,KM,w}\right)\cdot\alpha_1^{r,3,8}, \tag{30}$$

where $m_{E,KM,w}$ - elements of a subset $M_{E,KM,p}$; $n_{M_{E,KM,p}}$ – is the number of elements of the subset $M_{E,KM,p}$; $n_{1,S_i,3,8,k}$ – the number of computer stations in the network that are within the same firewall, that is, the implementation of the firewall corresponds to a subset $M_{E,KM,p}$; $n_{1,S_i,3,8,p}$ – number of computer stations in the corporate network.

The value of $\alpha'_{1,S_i,3,8}$, evaluated by formula (30) will be the same for those computer stations in the network that are within the same firewall. If there are several firewalls for different segments of the corporate network, then the values of $\alpha'_{1,S_i,3,8}$, will be different for them, respectively.

The use of intrusion detection systems (IDSs) in corporate networks is important in the context of ensuring security and, consequently, protecting information. Let's first consider IDSs located in the network nodes. Their functioning is based on the use of five main types of sensors, which we will denote as elements of the set $M_{H,KM}$: $m_{H,KM,1}$ – log analyzers; $m_{H,KM,2}$ – feature sensors; $m_{H,KM,3}$ – file integrity controllers; $m_{H,KM,4}$ – application behavior analyzers; $m_{H,KM,5}$ – system call analyzers. The number of sensors can be larger, so let $n_{M_{H,KM}}$ – be the number of elements of $M_{H,KM}$. In addition, these elements have different weights in the context of ensuring security in a corporate network. For example, log analyzers have to monitor events that may cause some security impact. However, they can only react to an event after it has occurred, not prevent it. Similarly, execution of the feature sensors designed to analyze traffic is performing during the attacks, and system call analyzers can prevent malicious actions. In a holistic context, all elements of the set $M_{H,KM}$ are important, but certain elements are more important. Different subsets can be formed from the elements of $M_{H,KM}$. Within certain segments, there may be different IDSs or the same or similarly configured IDSs. In addition, there may be IDSs in different physically distributed segments of the corporate network. In this situation such cases will be presented via the subsets union. This will affect the security of the corporate network. The security level value will be evaluated taking into account all available subsets. The local level of significance for the characteristic indicator of node's IDS in corporate networks will be the same for all nodes in the network part with the same implementation. Let's define a component $\alpha'_{1,S_i,3,9}$, that will form the value of $\alpha'_{1,S_i,3}$, and its value will be a characteristic indicator of node's IDSs in the corporate network. Let us define the value of $\alpha'_{1,S_i,3,9}$ for all computer stations so that it belongs to the interval $\left[1-\alpha_1^{r,3,9};1\right]$, where the local level of significance $\alpha_1^{r,3,9}$ is a fraction of one. Let's define the elements of $M_{H,KM}$, as the coordinates of vector $\left(m_{H,KM,1},m_{H,KM,2},m_{H,KM,3},m_{H,KM,4},m_{H,KM,5}\right)$, and assume that coordinates with a higher number will have more weight in the context of security. Let's introduce a function $f_{M_{H,KM}}$ whose value is given as follows:

$$f_{M_{H,KM}}\left(m_{E,KM,w}\right)=\frac{1}{w+1}, \tag{31}$$

where $w=1,2,\dots,n_{M_{E,KM}}$; $n_{M_{H,KM}}$ - is the number of elements of the set $M_{H,KM}$.
Let's evaluate the value of $\alpha'_{1,S_i,3,9}$ as follows:

$$\alpha'_{1,S_i,3,9}=1-\frac{n_{1,S_i,3,9,k}}{n_{1,S_i,3,9,p}}\cdot\prod_{w=1}^{n_{M_{H,KM,p}}}f_{M_{H,KM}}\left(m_{H,KM,w}\right)\cdot\alpha_1^{r,3,9} \tag{32}$$

where $m_{H,KM,w}$ - elements of a subset $M_{H,KM,p}$; $n_{M_{H,KM,p}}$ – number of elements of the subset $M_{H,KM,p}$; $n_{1,S_i,3,9,k}$ – the number of computer stations in the network that are within the same type of node's IDS, i.e., the implementation corresponds to a subset $M_{H,KM,p}$; $n_{1,S_i,3,9,p}$ – number of computer stations in the corporate network.
The value of $\alpha'_{1,S_i,3,9}$, evaluated by formula (32) will be the same for those computer stations in the network that are within the same type of node's IDS.

Let's take a look at network IDS. They can be different and this will affect their performance. For example, network IDS can be implemented in the network in places where you can monitor the traffic of all devices, or in a subnetwork with firewalls to protect them, or in a specially dedicated computer system. Network IDS perform the monitoring concerning the attack signs. The targeting and selection of such signs also affects their effectiveness. Therefore, taking into account the various configuration features of network IDS that will affect network security, we introduce a set $M_{N,KM}=\{m_{N,KM,1},m_{N,KM,2},\dots,m_{N,KM,n_{M_{N,KM}}}\}$, the elements of which will be configuration features, used feature sensors, and including of computer stations into the network. Let's assume that these elements have different weights in the context of ensuring security in a corporate network if they are applied in a certain part of it. Different subsets can be formed from the elements of the set $M_{N,KM}$ Also, within certain segments, there

may be different network IDSs, for example, in different geographically distributed segments of a corporate network, in this situation such cases will be defined by the subsets union. This will affect the security of the corporate network. The value of the security level will be evaluated taking into account all available subsets. The local level of significance for the characteristic indicator of network IDSs in corporate networks will be the same for all nodes in the network part with the same implementation. Let's introduce the component $\alpha'_{1,S_i,3,10}$, which will form the value $\alpha'_{1,S_i,3}$, and its value will be a characteristic indicator of network IDSs. The value $\alpha'_{1,S_i,3,10}$ will be determined for all computer stations so that it belongs to the interval $\left[1-\alpha_1^{r,3,10};1\right]$, where the local level of significance $\alpha_1^{r,3,10}$ is a fraction of one. Let's define the elements of the set $M_{N,KM}$, as the coordinates of the vector $\left(m_{N,KM,1}, m_{N,KM,2}, m_{N,KM,3}, m_{N,KM,4}, m_{N,KM,5}\right)$, and assume that coordinates with a higher number will have more weight in the context of security. Let's introduce a function $f_{M_{N,KM}}$ whose value is given as follows:

$$f_{M_{N,KM}}\left(m_{N,KM,w}\right)=\frac{1}{w+1}, \quad (33)$$

where $w=1,2,\dots,n_{M_{E,KM}}$; $n_{M_{N,KM}}$ - number of elements of the set $M_{N,KM}$.

For network IDSs, the evaluation of the $\alpha'_{1,S_i,3,10}$ value is as follows:

$$\alpha'_{1,S_i,3,10}=1-\frac{n_{1,S_i,3,10,k}}{n_{1,S_i,3,10,p}}\cdot\prod_{w=1}^{n_{M_{N,KM,p}}} f_{M_{N,KM}}\left(m_{N,KM,w}\right)\cdot\alpha_1^{r,3,10}, \quad (34)$$

where $m_{N,KM,w}$ - elements of a subset $M_{N,KM,p}$; $n_{M_{N,KM,p}}$ – number of elements of the subset $M_{N,KM,p}$; $n_{1,S_i,3,10,k}$ –the number of computer stations in the network that are within the same type of network IDSs configuration, i.e., the implementation corresponds to a subset $M_{N,KM,p}$; $n_{1,S_i,3,10,p}$ – number of computer stations in the corporate network.

The distribution of computer stations and servers in the corporate network into zones with different security levels will affect the value of the security levels in each zone. For example, some computer stations may be located in a segment or zone where increased security measures are implemented for users, while some users may be located in a zone where, for example, there is access to the Internet via Wi-Fi, the ability to use computers without dividing users into target groups. In addition, the company's computer network may not be divided into zones with different security levels, and all nodes will be in the same zone. Let's introduce the component $\alpha'_{1,S_i,3,11}$, which will form the value of $\alpha'_{1,S_i,3}$, and its value will be a characteristic indicator of the division of nodes in the network between zones with different security levels. The value of $\alpha'_{1,S_i,3,11}$ is defined for all computer stations so that it belongs to the interval $\left[1-\alpha_1^{r,3,11};1\right]$, where the local level of significance $\alpha_1^{r,3,11}$ is a fraction of one. The evaluation of the value $\alpha'_{1,S_i,3,11}$ is as follows:

$$\alpha'_{1,S_i,3,11}=1-\frac{n_{1,S_i,3,11,k}}{n_{1,S_i,3,11,p}}\cdot\alpha_1^{r,3,11}, \quad (35)$$

where $n_{1,S_i,3,11,k}$ – the number of computer stations in the network that are outside the high security zone; $n_{1,S_i,3,11,p}$ – number of computer stations in the corporate network.

If there are several zones in the network with different security levels, then one network segment is selected as the zone of increased security. All other segments are considered to be outside the zone of increased security.

Antivirus software must be used in a corporate network to ensure security and protection of information. As a rule, such tools are used to create a multi-level system, in which each level has its own purpose and focus. Therefore, let's introduce a set $M_{AZ,KM}=\{m_{AZ,KM,1}, m_{AZ,KM,2},\dots,m_{AZ,KM,n_{M_{AZ,KM}}}\}$, the elements of which will be system's levels with the corresponding specialized antivirus tools in the network. The number of elements of $M_{AZ,KM}$ is denoted by $n_{M_{AZ,KM}}$. For example, the first level can be used to scan network traffic, the second to protect mail servers, the third to protect file servers, the fourth to scan files on a particular computer station, and the fifth to monitor events on a particular computer station. Let's assume that these elements are of equal importance in the context of antivirus protection in a corporate network. In addition, they can be applied to specific network segments. In certain network segments, all levels can be applied, while in others, only some can be applied. Different subsets can be formed from the elements of the set $M_{AZ,KM}$. Combining several elements of the set $M_{AZ,KM}$ into subsets improves the security level in network nodes. Within certain segments, there may be different levels, so such cases will be defined by combining subsets. This will affect the antivirus protection of the corporate network. The security level value should take into account all available subsets. The local significance level for the characteristic indicator of multi-level antivirus protection in corporate networks may be different for all nodes in the network. It depends on the specific antivirus software of the computer station. Let's define a compound part $\alpha'_{1,S_i,3,12}$, which will form the value of $\alpha'_{1,S_i,3}$, and its value will be a characteristic indicator of multi-level antivirus protection in corporate networks. We define the value of $\alpha'_{1,S_i,3,12}$ for all computer stations so that it belongs to the interval $\left[1-\alpha_1^{r,3,12};1\right]$, where the local level of significance $\alpha_1^{r,3,12}$ is a fraction of one. Let's introduce a function $f_{M_{AZ,KM}}$ whose value is given as follows:

$$f_{M_{AZ,KM}}\left(m_{AZ,KM,w}\right)=\begin{cases}0, if\, n_{M_{AZ,KM}}=0;\\ \frac{1}{n_{M_{AZ,KM}}}, if\, n_{M_{AZ,KM}}>0,\end{cases}, \quad (36)$$

where $w=1,2,\dots,n_{M_{AZ,KM}}$; $n_{M_{AZ,KM}}$ - number of elements of the set $M_{AZ,KM}$.

Let's evaluate the value of $\alpha'_{1,S_i,3,12}$ for each i-th node as follows:

$$\alpha'_{1,S_i,3,12}=$$
$$=1-\frac{n_{1,S_i,3,12,k}}{n_{1,S_i,3,12,p}}\cdot\prod_{w=1}^{n_{M_{AZ,KM,p}}} f_{M_{AZ,KM}}\left(m_{AZ,KM,w}\right)\cdot\alpha_1^{r,3,12}, \quad (41)$$

where $m_{AZ,KM,w}$ - elements of a subset $M_{AZ,KM,p}$; $n_{M_{AZ,KM,p}}$ – number of elements of the subset $M_{AZ,KM,p}$; $n_{1,S_i,3,12,k}$ – the number of computer stations in the network that use the same type of anti-virus protection system, that is, the implementation

corresponds to a subset $M_{AZ,KM,p}$; $n_{1,S_i,3,12,p}$ – number of computer stations in the corporate network.

The value of the security level in a computer network node is also influenced by the load characteristics and the available computing resources of the computer station. If the resources of the computer station do not meet the specified load parameters, then the execution of tasks in it slows down and additional processing of events regarding abnormal or malicious actions will become more difficult. Such characteristic indicators are the amount of free space on the hard disk, RAM, the use of virtual memory technology, etc. For these characteristic indicators, we keep local significance levels $\alpha_1^{r,3,13}$, $\alpha_1^{r,3,14}$, $\alpha_1^{r,3,15}$ respectively. The values of $\alpha'_{1,S_i,3,13}$, $\alpha'_{1,S_i,3,14}$, $\alpha'_{1,S_i,3,15}$ respectively, are defined for all computer stations in the network so that they belong to the intervals $\left[1-\alpha_1^{r,3,13};1\right]$, $\left[1-\alpha_1^{r,3,14};1\right]$, $\left[1-\alpha_1^{r,3,15};1\right]$ and are evaluated for each i-th node as follows:

$$\alpha'_{1,S_i,3,13}=1-\frac{u_{1,S_i,3,13,k}}{u_{1,S_i,3,13,p}}\cdot\alpha_1^{r,3,13}, \quad (38)$$

$$\alpha'_{1,S_i,3,14}=1-\frac{u_{1,S_i,3,14,k}}{u_{1,S_i,3,14,p}}\cdot\alpha_1^{r,3,14}, \quad (39)$$

$$\alpha'_{1,S_i,3,15}=1-\frac{u_{1,S_i,3,15,p}}{u_{1,S_i,3,15,p}+u_{1,S_i,3,15,k}}\cdot\alpha_1^{r,3,15}, \quad (40)$$

where $u_{1,S_i,3,13,k}$ – is the amount of hard disk space used by the i-th node in the network; $u_{1,S_i,3,13,p}$ – is the total amount of hard disk space of the i-th node in the network; $u_{1,S_i,3,14,k}$ – is the amount of filled space of the RAM of the i-th node in the network; $u_{1,S_i,3,14,p}$ – is the total amount of RAM space of the i-th node in the network; $u_{1,S_i,3,15,k}$ – is the amount of hard disk space occupied by the process, which is implemented according to the virtual memory technology, of the i-th node in the network; $u_{1,S_i,3,15,p}$ – is the total amount of RAM and hard disk space occupied by the process, which is implemented according to the virtual memory technology, of the i-th node in the network.

The computing resources of computer stations in terms of their utilization rate are important and affect the values of security levels. If the corporate network does not have firewalls, antivirus protection systems and intrusion detection systems, then the values corresponding to the confidence levels in the evaluation result are set equal to the lower limits of the respective intervals.

Similarly, other characteristic indicators can be added to the defined characteristic indicators that will form the value of the security level at a particular node in the network, i.e. their number can be increased.

The value of $\alpha'_{1,S_i,3}$, which takes into account the security level of a node in the network, is evaluated taking into account its component values $\alpha'_{1,S_i,3,1}-\alpha'_{1,S_i,3,15}$ as follows:

$$\alpha'_{1,S_i,3}=\frac{1}{15}\cdot\sum_{w=1}^{15}\alpha'_{1,S_i,3,w}, \quad (41)$$

The value will be in the interval $\alpha'_{1,S_i,3}$ $\left[1-\alpha_1^{r,3};1\right]$, where $\alpha_1^{r,3}$ is the significance level and is evaluated as follows: $\alpha_1^{r,3}=\frac{1}{15}\cdot\sum_{w=1}^{15}\alpha_1^{r,3,w}$.

In the system S at a certain time there will be components that may have a system decision center, but they will be inactive, then the values of the security level in such components will differ from the values in the components in which the decision center is active. Dynamic components may use certain functions related to ensuring the functioning of the system's decision center, but the component itself will not be an active part of the decision center at the current time. The value of $\alpha'_{1,S_i,4}$, which takes into account the execution of functions in components with inactive subsystems of the system's decision center and belongs to the interval $\left[1-\alpha_1^{r,4};1\right]$, is evaluated as follows:

$$\alpha'_{1,S_i,4}=1-\frac{n_{1,S_i,4,p}}{n_{1,S_i,4,k}}\cdot\alpha_1^{r,4}, \quad (42)$$

where $n_{1,S_i,4,k}$ - is the total number of functions in the i-th component; $n_{1,S_i,4,p}$ - is the number of functions in the i-th component taken for its formation from the remaining components of the system S; $\alpha_1^{r,4}$ – is the level of significance for the i-th component with functions of inactive subsystems of the system decision center.

The number of functions that participated in the current task related to the tasks of the decision center of the system S, will affect the value of the security level, because when performing more functions in the implementation of distributed computing, the confidence in their results will be less than when performing fewer functions. The value of $\alpha'_{1,S_i,5}$ belongs to the interval $\left[1-\alpha_1^{r,5};1\right]$ and will be evaluated as follows:

$$\alpha'_{1,S_i,5}=1-\frac{n_{1,S_i,5,p}}{n_{1,S_i,5,k}}\cdot\alpha_1^{r,5}, \quad (43)$$

where $n_{1,S_i,5,p}$ - the number of functions in the i-th component that were involved in the current task, which is related to the tasks of the system decision center S; $n_{1,S_i,5,k}$ - the largest number of functions that were involved in the current task, which is related to the tasks of the decision-making center of the system S, i.e. $n_{1,S_i,5,k}=max\left(n_{1,S_1,5,p},n_{1,S_2,5,p},\dots,n_{1,S_k,5,p}\right)$, $i=1,2,\dots,k$; $\alpha_1^{r,5}$ – is the level of significance for the i -th component in terms of taking into account the number of functions that participated in the current task related to the tasks of the decision-making center of the system S.

Let us consider the computations performed by the functions that are not the functions of the decision center of the system S. These are computations related to the system's tasks of detecting malicious events or anomalies and are not related to computations performed for the tasks of the decision center. The functions for performing such computations may be resident or called by other functions. Since they will not directly affect decision-making and there may be no signs of malicious influences or anomalies in a particular node in the network, the security level for such computations will be higher than in the case of computations for the decision center of the system. These computations will also be mostly distributed, and therefore will require a certain amount of time to execute and generate the result.

In this regard, the value of the results of such calculations can also be different depending on:

1. the order of execution of functions-subsets in dynamic components,

2. time spent on forwarding execution results,
3. node security level,
4. performance of subset functions in three types of components (with active existing subsystems of the system decision-making center, with inactive existing subsystems of the system decision-making center, with missing subsystems of the system decision-making center),
5. the number of functions that were used in the execution of the task.

Then, the weights of the functions are defined as functions with five arguments as follows:

$$\alpha'_{2,S_{k+1,n}} = \\ = f_{\alpha'_{2,S_{k+1,n}}}\left(\alpha'_{2,S_{k+1,n},1}, \alpha'_{2,S_{k+1,n},2}, \alpha'_{2,S_{k+1,n},3}, \alpha'_{2,S_{k+1,n},4}, \alpha'_{2,S_{k+1,n},5}\right) \quad (44)$$

where $S_{k+1,n}$ – system components that has no functions for the system decision center;) $(k+1)$- n – numbers of components that do not have a center; $\alpha'_{2,S_{k+1,n},1}$ - a value that takes into account the order of execution of functions in dynamic components; $\alpha'_{2,S_{k+1,n},2}$ - is a value that takes into account the relative time spent on sending intermediate results of the performed evaluations; $\alpha'_{2,S_{k+1,n},3}$ - a value that takes into account the security level of the node in the network; $\alpha'_{2,S_{k+1,n},4}$ - a value that takes into account the execution of functions in three types of components (with active existing subsystems of the system's decision center, with inactive existing subsystems of the system's decision center, with missing subsystems of the system's decision center); $\alpha'_{2,S_{k+1,n},5}$ - a value that takes into account the number of functions that participated in the task.

The values of $\alpha'_{2,S_{k+1,n},1}, \alpha'_{2,S_{k+1,n},2}, \alpha'_{2,S_{k+1,n},3}, \alpha'_{2,S_{k+1,n},4}, \alpha'_{2,S_{k+1,n},5}$ belong to the intervals $[1-\alpha_2^{r,w};1]$ $(w=1,2,\dots,5)$, in which the levels of significance $\alpha_2^{r,w}$ $(w=1,2,\dots,5)$ for each of them are set accordingly. We will evaluate the values taking into account their correlation with the values of the arguments of the function $f_{\alpha'_{2,S_i}}$, since some of the characteristic indicators will be the same for each of them.

The values of $\alpha'_{2,S_{k+1,n},1}$ will depend on the number of functions that will form the functionality of the system component compared to the number of functions that are intended to ensure the functioning of the system decision-making center. Therefore, we will evaluate it taking into account the value of $\alpha'_{1,S_i,j,1}$ as follows:

$$\alpha'_{2,S_{k+1,n},j,1} = 1-\alpha_2^{r,1}+\left(n_{S_{k+1,n},max,f,2}-f_{nom}(j,1)\right)\cdot\frac{\alpha_2^{r,1}}{n_{S_{k+1,n},max,f,2}-1}\frac{-K_{f,j}\cdot\alpha_2^{r,1}}{n_{S_{k+1,n},max,f,2}}; \alpha_2^{r,1} = \frac{n_{2,S_{k+1,n},j,1,k}}{n_{2,S_{k+1,n},j,1,p}}\cdot\alpha_1^{r,1}, \quad (45)$$

where $n_{2,S_{k+1,n},1,p}$ - is the number of functions that will form the functionality of the i-th component; $i=k+1,k+2,\dots,n$; $n_{2,S_{k+1,n},1,k}$ - number of functions that are not used to form the decision-making center of the system S; $f_{nom}(j,1)$ – is the first smallest value of the coordinate of the vector $v_{\alpha'_{1,S_{k+1,n}},j}$; $n_{S_{k+1,n},max,f,2}$ - is the number of functions involved in solving a particular task; $K_{f,j}$ – is the number of calls to the j-th function; the vector $v_{\alpha'_{1,S_{k+1,n}},j}$ is formed similarly to the vector $v_{\alpha'_{1,S_i},j}$ by the same constructive rule.

The value of $\alpha'_{2,S_{k+1,n},1}$ according to formula (45) is defined as follows:

$$\alpha'_{2,S_{k+1,n},1} = \frac{\sum_{j=1}^{n_{S_{k+1,n},max}}\alpha'_{2,S_{k+1,n},j,i}}{n_{S_{k+1,n},max}}, \quad (46)$$

where $n_{S_{k+1,n},max}$ - is the number of functions involved in solving a particular task.

Thus, increasing the number of functions in the components with that are not used to form the decision-making center of the system S, will bring the value of the significance level $\alpha_2^{r,1}$ closer to the value of $\alpha_1^{r,1}$. The value of the significance level $\alpha_2^{r,1}$ is less than the value of $\alpha_1^{r,1}$, so the interval in which the value of the confidence level in the result will be located will be smaller and, accordingly, will be closer to the value of one.

Similarly, we will define the value of $\alpha'_{2,S_{k+1,n},2}$, which takes into account the relative time spent on sending intermediate results of the performed evaluations, as the value of $\alpha'_{2,S_i,2}$ and the level of significance $\alpha_2^{r,2}$ as follows:

$$\alpha'_{2,S_{k+1,n},j,2} = 1-\frac{\sum_{q=1}^{K_{f,j,t}} f_{nom,t}(j,q)}{\sum_{j=1}^{n_{S_{k+1,n},max}}\sum_{q=1}^{K_{f,j,t}} f_{nom,t}(j,q)}\cdot\alpha_2^{r,2};$$

$$\alpha_2^{r,1} = \frac{n_{2,S_{k+1,n},1,k}}{n_{2,S_{k+1,n},1,p}}\cdot\alpha_1^{r,1}, \quad (47)$$

where $n_{2,S_{k+1,n},1,p}$ - is the number of functions that will form the functionality of the i-th component; $n_{2,S_{k+1,n},1,k}$ - number of functions that are not used to form the decision-making center of the system S; $i=k+1,k+2,\dots,n$; $f_{nom,t}(j,q)$ – is the value of the coordinate of the vector $v_{\alpha'_{1,S_{k+1,n}},j,t}$, which are formed in the vector similarly to the constructive rule for forming the coordinates of the vector $v_{\alpha'_{1,S_i},j,t}$; $n_{S_{k+1,n},max}$ - the number of functions in the component; $K_{f,j,t}$ – is the number of calls to the j-th function.

The value of $\alpha'_{1,S_i,2}$ is defined as follows:

$$\alpha'_{2,S_{k+1,n},2} = \frac{\sum_{j=1}^{n_{S_{k+1,n},max}}\alpha'_{2,S_i,j,i}}{n_{S_{k+1,n},max}}, \quad (48)$$

where $n_{S_{k+1,n},max}$ - the number of functions in the component.

The value of $\alpha'_{2,S_{k+1,n},3}$, which takes into account the security level of the node in the network, is determined similarly to the definition of the value of $\alpha'_{1,S_i,3}$. At the same time, we will supplement it with a characteristic indicator that will reflect the activity of malicious sign detectors. The value of the significance level $\alpha_2^{r,3}$ will be correlated with the value of $\alpha_1^{r,3}$ and determined according to the local significance levels $\alpha_2^{r,3,w}$ $(w=1,2,\dots,16)$. We will evaluate the value as follows:

$$\alpha'_{2,S_{k+1,n},3} = \frac{1}{16}\cdot\sum_{w=1}^{16}\alpha'_{2,S_{k+1,n},3,w},$$

$$\alpha_2^{r,3} = \frac{1}{16}\cdot\sum_{w=1}^{16}\alpha_2^{r,3,w},$$

$$\alpha_2^{r,3,w} = \frac{n_{2,S_{k+1,n},1,k}}{n_{2,S_{k+1,n},1,p}} \cdot \alpha_1^{r,3,w}; \quad w = 1,2,\dots,16, \qquad (49)$$

where $n_{2,S_{k+1,n},1,p}$ - is the number of functions that will form the functionality of the i-th component; $n_{2,S_{k+1,n},1,k}$ - number of functions that are not used to form the decision-making center of the system S; $i = k+1, k+2, \dots, n$.

To evaluate the value of $\alpha'_{2,S_{k+1,n},3}$ you need to determine the value of $\alpha'_{2,S_{k+1,n},3,16}$ of the characteristic indicator that will reflect the activity of malicious sign detectors. A particular i-th computer station has a subsystem for monitoring malicious events, the number of such subsystem tools and the number of active ones. It will indicate the value of the security level taking into account this component and its task. We will determine its value in the interval $\left[1-\alpha_2^{r,3,16}; 1\right]$ as follows:

$$\alpha'_{2,S_{k+1,n},3,16} = 1 - \frac{n_{2,S_{k+1,n},1,k,16}}{n_{2,S_{k+1,n},1,p,16}} \cdot \alpha_2^{r,3,16}, \qquad (50)$$

where $n_{2,S_{k+1,n},1,p,16}$ – is the total number of feature sensors in the i-th component; $n_{2,S_{k+1,n},1,k,16}$ - is the number of feature sensors in the i-th component activated by malicious actions or abnormal behavior; $\alpha_2^{r,3,16}$ – local level of significance.

The value of $\alpha'_{2,S_{k+1,n},j,4}$ takes into account the execution of functions in three types of components (with active existing subsystems of the system decision center, with inactive existing subsystems of the system decision center, with missing subsystems of the system decision center). Let us evaluate its value taking into account the membership of the interval $\left[1-\alpha_2^{r,4}; 1\right]$ as follows:

$$\alpha'_{2,S_{k+1,n},4} = 1 - \left(\frac{n_{1,S_{k+1,n},4,p,1}}{n_{1,S_{k+1,n},4,k}} + \frac{n_{1,S_{k+1,n},4,p,2}}{2 \cdot n_{1,S_{k+1,n},4,k}} + \frac{n_{1,S_{k+1,n},4,p,3}}{3 \cdot n_{1,S_{k+1,n},4,k}}\right) \cdot \alpha_2^{r,4};$$
$$\alpha_2^{r,4} = \frac{n_{2,S_{k+1,n},1,p}}{n_{2,S_{k+1,n},1,k}} \cdot \alpha_1^{r,4}, \qquad (51)$$

where $n_{1,S_{k+1,n},4,p,1}$ - the number of functions in the i -th component taken for its formation from the active subsystems of the decision-making center of the system components S; $n_{1,S_{k+1,n},4,p,2}$ - number of functions in the i – th component taken for its formation from the inactive existing subsystems of the decision-making center of the system components S; $n_{1,S_{k+1,n},4,p,3}$ - number of functions in the i – th component taken for its formation with the missing subsystems of the decision-making center of the system components of the system S; $n_{1,S_{k+1,n},4,k}$ - is the total number of functions in the i–th component; $n_{1,S_{k+1,n},4,p}$ - is the number of functions in the i–th component taken for its formation from the rest of the system components S; $\alpha_2^{r,4}$ – is the level of significance for the i –th component; $i = k+1, k+2, \dots, n$.

The value of $\alpha'_{2,S_{k+1,n},5}$ takes into account the number of functions that participated in the task. We define it to be in the interval $\left[1-\alpha_2^{r,5}; 1\right]$, and evaluate its value as follows:

$$\alpha'_{2,S_{k+1,n},5} = 1 - \frac{n_{2,S_{k+1,n},5,p}}{n_{2,S_{k+1,n},5,k}} \cdot \alpha_2^{r,5}; \ \alpha_2^{r,4} = \frac{n_{2,S_{k+1,n},1,p}}{n_{2,S_{k+1,n},1,k}} \cdot \alpha_1^{r,4}, \qquad (52)$$

where $n_{2,S_{k+1,n},5,p}$ - is the number of functions in the i–th component that were involved in the current task; $n_{2,S_{k+1,n},5,k}$ - the largest number of functions that were involved in the current task; $\alpha_2^{r,5}$ – is the level of significance for the i –th component, taking into account the number of functions that participated in the current task.

Thus, the values of the characteristic indicators of the components of the system S are formed for the case when the functions perform tasks that do not belong to the tasks of the decision-making center.

Let's consider distributed computing, which is performed by functions. This option involves the execution of functions that do not belong to the decision center. But after the execution of functions they call the functions of the decision center for additional computations. As a result, the functions of the decision center and the functions not related to the decision center will be executed. Similarly to the previous two cases, we will define the weights of the functions and define them as functions with five arguments as follows:

$$\alpha'_{3,S_{1,n}} = f_{\alpha'_{3,S_{1,n}}}\left(\alpha'_{3,S_{1,n},1}, \alpha'_{3,S_{1,n},2}, \alpha'_{3,S_{1,n},3}, \alpha'_{3,S_{1,n},4}, \alpha'_{3,S_{1,n},5}\right), \qquad (53)$$

where $S_{1,n}$ – system components that contain functions of all types; $i = 1,2,\dots,n$; i – number of component $\alpha'_{3,S_{1,n},1}$ - a value that takes into account the order of execution of functions in dynamic components; $\alpha'_{2,S_{1,n},2}$ - the evaluation takes into account the relative time spent on sending intermediate results of the performed evaluations; $\alpha'_{2,S_{1,n},3}$ - a value that takes into account the security level of the node in the network; $\alpha'_{2,S_{1,n},4}$ - a value that takes into account the execution of functions in three types of components (with active existing subsystems of the system's decision center, with inactive existing subsystems of the system's decision center, with missing subsystems of the system's decision center); $\alpha'_{2,S_{1,n},5}$ - a value that takes into account the number of functions that participated in the task.

The values of $\alpha'_{3,S_{1,n},1}, \alpha'_{3,S_{1,n},2}, \alpha'_{3,S_{1,n},3}, \alpha'_{3,S_{1,n},4}, \alpha'_{3,S_{1,n},5}$ belong to the intervals $[1-\alpha_3^{r,w}; 1]$ $(w = 1,2,\dots,5)$, in which the levels of significance $\alpha_3^{r,w}$ $(w = 1,2,\dots,5)$ for each of them are set accordingly. We will evaluate the values taking into account their correlation with the values of the arguments of the function $f_{\alpha'_{3,S_i}}$, since some of the characteristic indicators will be the same for each of them. Given the coverage of all functions by these characteristics and the processes that will begin in the functions used to detect malicious actions and anomalies and continue with urgent calls to the functions related to the formation of the decision-making center, the complexity and scale of such distributed computing will be greater, as well as the possible reduction in the value of the security level in certain components. Therefore, we will evaluate the value of $\alpha'_{3,S_{1,n},1}, \alpha'_{3,S_{1,n},2}, \alpha'_{3,S_{1,n},3}, \alpha'_{3,S_{1,n},4}, \alpha'_{3,S_{1,n},5}$ taking into account the values for the first and second cases and the formulas for evaluation their values as follows:

$$\alpha_3^{r,w} = \alpha_1^{r,w} + \alpha_2^{r,w}; \ \alpha'_{3,S_{1,n},j,w} = \alpha'_{1,S_i,j,w} + \alpha'_{2,S_{k+1,n},j,w} - 1; \quad w = 1,2,\dots,5, \qquad (54)$$

The values of the characteristic indicators can be refined in the course of detailing with the involvement of new parameters. Their number can also be expanded. A large number of such

characteristic indicators will require the application of data dimensionality reduction methods, such as the principal component method. As a result, it was found that the values of security levels are convergent regardless of the number of system components S, i.e., the number of computer stations in the corporate network. Also, all the obtained values belong to the specified intervals. According to the graphs, it was found that the values of the security levels of the system components will be higher with the minimum number of active components in which the decision-making center of the system is located.

Thus, the obtained analytical expressions for the characteristic indicators of the values of the security levels of components are mathematical models that formalize the architecture of the components of the system S according to their functions, their purpose, interaction, place of execution, formation of a decision-making center and assessment of the security level of the computations performed. Taking into account the specific purpose of the system S the processes at the level of functions in it and components are set by mathematical models of the characteristic indicators of the security levels of components. The values of the characteristic indicators of the security levels of the components will serve as the basis for the formation of decisions of the system S regarding its further steps and the identification of malicious software.

V. EXPERIMENTS

The purpose of experiments concerning developed partially centralized system is to determine the following indicators:

1. the degree of system sustainability;
2. the degree of system's degradation and its components' degradation;
3. the relevance degree of time task execution and time response to events in the system and computer stations;
4. the detection accuracy of the malicious software.

Let us consider determining the degree of sustainability of a system during its functioning, taking into account the specifics of the tasks it performs. We will study the sustainability of the system in the context of its ability to continue its functioning and executing its tasks under the of changes in the operating environment caused by internal processes of the system itself and external processes that can be caused by various reasons, including malware, with minimal change or loss of its functionality. Let us consider the stable system state, that includes installed system components, if the system is functioning without any external influences, that may cause the change of the functioning reliability of computer stations in the corporate network.

The remaining states of the system S will be classified as unstable. Among the stable states we will distinguish the partial stable states, which include states that activate methods of detecting malware in a computer network. Then, the system can be in three states and the transitions between the states can be represented by a complete graph. The stable state of the system is caused by the absence of disturbing factors. The partial stable state is caused by the absence of disturbing factors and, at the same time, the activation of the subsystem to detect malware. The sustainability of a system S will be characterized by its ability to return to the stable state after being in a partial stable state or unstable state. A system is considered unstable if it is in partial stable or in unstable state for a long time and is unable to other states.

Let us consider the sustainability conditions of the system S. For each initial value that will be processed, the system must generate a result that will not leave it either in stable state or in a partial stable state. If the system S does not receive input values for a certain period of time, then it does not generate any decisions.

Thus, the given conditions of system sustainability coincide with the given conditions of its functioning in computer networks according to the method of organizing the functioning of partially distributed systems.

Then, such conditions will be considered to correspond to the internal principles of the system's functioning and their observance and analysis can be the basis for studying the system's sustainability in terms of its stable functioning. The sustainability indicator will be set for a specific impact characteristic. The system S will be characterized by dynamic sustainability, which reflects the ability to restore the initial state after the impact of factors. The system S, due to the presence of different component states, will have a large number of variants for component formation, so we will consider it as a nonlinear dynamic system.

The system S will be considered as a self-organized discrete system, since it will be in states depending on the states of its components. Taking into account the time the transition between states will be characterized by the breaking point of the first kind. For example, let's show the dependency graph of component states on time in Fig. 3.

In Fig. 4 the time chart for a component in certain states is presented. It demonstrates states which are indicated by vertices, and transitions between them by arcs. The transitions between vertices 1 and 2, 2 and 3, 4 and 5, 6 and 7, 8 and 9, 10 and 11 are actually a presentation of the component being in the current state for a certain time.

The presentation of the component staying in the same state is related to the periodic accounting of the component's state over time. It is reflected in the timeline. The breaking points of the first kind are presented on the graph by the transitions between vertices 3 and 4, 5 and 6, 7 and 8, 9 and 10.

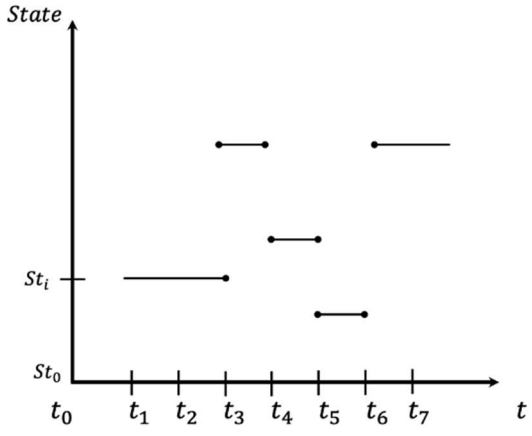

Figure 3. Dependency graph of component states on time.

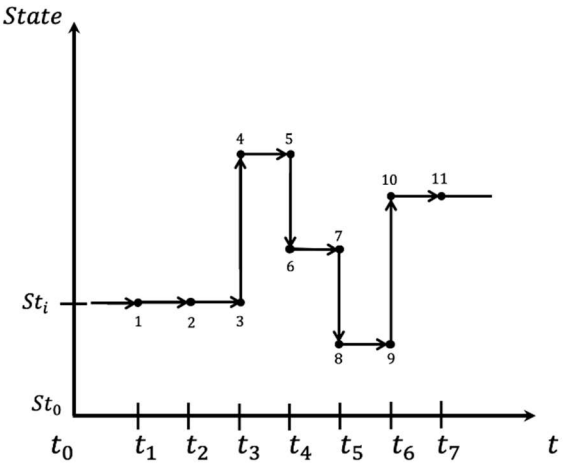

Figure 4. Time chart for a component in certain states.

The discreteness of system component is approved by the presence of breakpoints during a certain time of functioning. Since discreteness is not caused by time periods, but by the presence of components in different states depending on time and cannot be set continuously, the system S will be discrete in terms of level and time. In terms of time, the system will be discrete, because its functioning in computer stations will occur at the level of process execution. For example, let's demonstrate a fragment of the system functioning through quantization by level and time in Fig. 5.

Thus, the system S is discrete, and to determine the level of its sustainability, we consider the values of the characteristic indicators that allow us to obtain the values of the state of the components. These values are the arguments of the functions in formulas (12), (44), (53) and the corresponding analytical expressions are defined to calculate their values. For an analysis of the characteristic indicators values concerning the sustainability of the system S, we first consider two indicators, and then scale the result to all 15 indicators.

In order to investigate the system sustainability S let us use the values safety levels of the components. They are given by the set $B = \{\beta'_1, \beta'_2, \dots, \beta'_{N_B}\}$, where β'_i - is the value of the safety levels of the system components, N_B is the number of characteristic indicators, $i = 1,2, \dots, N_B$.

For each component of the system S we will introduce subsets $B_j = \{\beta'_{1,j}, \beta'_{2,j}, \dots, \beta'_{N_B,j}\}$ according to the given set B, the elements of which will be used to calculate the security level of the entire system.

The values $\beta'_{i,j}$ $(i = 1,2, \dots, N_B)$ will determine the confidence level in the results of distributed computing performed in different components of the system and characterize different indicators of security levels. For the values of $\beta'_{i,j}$ $(i = 1,2, \dots, N_B;\ j = 1,2, \dots, N;\ N$ – the number of components in the system that are installed in computer stations in the network), let's introduce a range in which the lower bound will be adjusted depending on the significance level parameter $\alpha_z^{r,i}$ $(i = 1,2, \dots, N_B;\ z = 1,2, \dots, N_z;\ N_z$ – the number of variants of function interaction of as follows: $[1 - \alpha_z^{r,i}; 1]$.

The level of significance is a fraction of one, which reflects the deviation from the level of confidence in the result of distributed computing due to some events, architectural features of the component, etc. Then, for two characteristic indicators of one component, for example, $\beta'_{1,j}, \beta'_{2,j}$ for the j–th component, the results of calculations can be presented on the coordinate plane by two points.

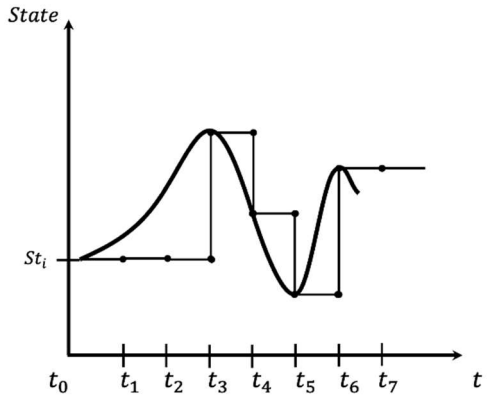

Figure 5. Graph of quantization by level and time.

If the system component is functioning stably, then the values of the points will be contained in a rectangle defined by the abscissa axis by segment $[1 - \alpha_z^{r,1}; 1]$ and the ordinate axis by segment $[1 - \alpha_z^{r,2}; 1]$. The presentation of the rectangle is shown in Fig. 6.

The same values of characteristic indicators can be presented on a graph of their values as a function of time (Fig. 7).

The results given in the graphs can correspond to the same characteristics as for the component states, as it is shown in Figs. 3- 5. This presentation of the values of the characteristic indicators confirms the discreteness of the system and makes it possible to determine the sustainability of the system using the values of the characteristic indicators.

The structural scheme of the system S is presented in Fig.8.

Fig. 9 presents , the designation B means a set of the values safety levels of the component, which is constantly updated at certain intervals, and the designation SDMC - means the decision-making center of the system S. The definition of the elements of the set is carried out linearly, so this part can be expressed by linear functions, but the second part of the system (system decision-making center) is nonlinear. The second part of system reflects the functioning of the system to perform specialized tasks and is nonlinear.

Let's set the elements of the set B to the coordinates of the vector. As a result, we get a state space with different vectors and their values. Let us present the refined structural diagram of the system S in Fig. 9.

Let us define the function $W_{S,c}^1$ to describe the decision center of the system as follows:

$$W_{S,c}^1 = \sqrt{\sum_{i=1}^{N_B} \sum_{j=1}^{N} {\beta'_{i,j}}^2}, \qquad (55)$$

where N_B – number of characteristic indicators, $i = 1,2, \dots, N_B$; $\beta'_{i,j}$ $(i = 1,2, \dots, N_B)$ values that will determine the level of confidence in the results of distributed computing performed in different components of the system and characterize different indicators of security levels; $j = 1,2, \dots, N$; N – the number of components in the system that are installed in computer stations in the network.

The value of the function $W_{S,c}^1$ will be a segment whose length does not exceed the value of $\sqrt{N_B \cdot N}$, and will characterize the state of the system when all computer stations are turned on and all components of the system S are active. The lower bound on the value of the function $W_{S,c}^1$ is $\sqrt{N \cdot \sum_{i=1}^{N_B} \left(1 - \alpha_z^{r,i}\right)^2}$. The geometric interpretation of the

function $W^1_{S,c}$ is a point in $N_B \cdot N$ – space with the number of coordinates $N_B \cdot N$. Therefore, the sustainability of the system S will depend on the value of the function $W^1_{S,c}$. If the value exceeds the value of $\sqrt{N_B \cdot N}$, the system will enter the unstable state and will remove components from its architecture that have the greatest impact on the value of the function $W^1_{S,c}$. After such component removals, the system S will return to the stable state and try to add components again step by step. If the values of some components are zero due to their absence in the system (computer stations are turned off), the value of the function $W^1_{S,c}$ is calculated for the available components and, then, the point is set in a space smaller than $N_B \cdot N$. In this case, the calculated value of the function $W^1_{S,c}$ will also be in the same interval. If the value of the function $W^1_{S,c}$ is less than the number $\sqrt{N \cdot \sum_{i=1}^{N_B}\left(1-\alpha_z^{r,i}\right)^2}$, then the system will also enter the unstable state. This state will be caused by the values safety levels of the components. Evaluated values did not belong to at least one of the intervals $\left[1-\alpha_z^{r,i};1\right]$, and was such smaller values than $1-\alpha_z^{r,i}$, that they have affected the overall resultant indicator. As a result these components are to be removed from the system's architecture. However, it may be that this value does not affect the overall indicator, although it is less than the set value. In this case the system will remain in the stable state and decide whether to analyze the value from the component and, if necessary, remove it from the system architecture.

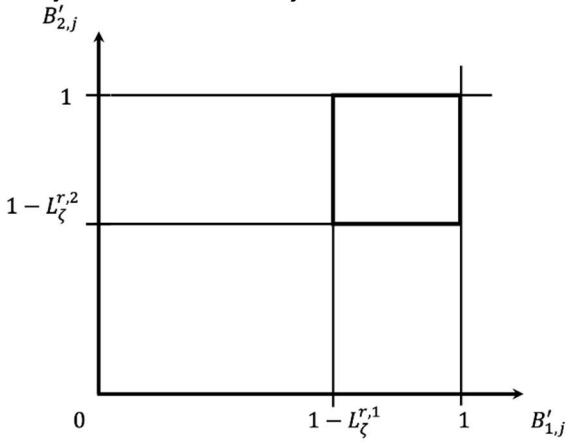

Figure 6. The range of values for two characteristic indicators.

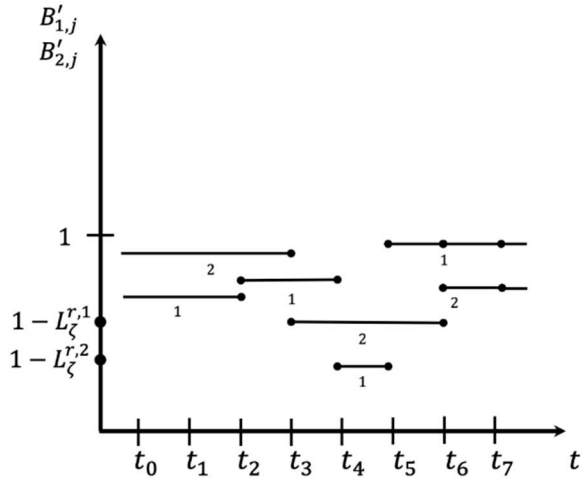

Figure 7. Graphs of dependence for two values of characteristic indicators on time.

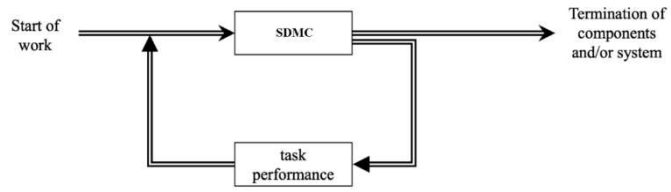

Figure 8. The structural scheme of the system S.

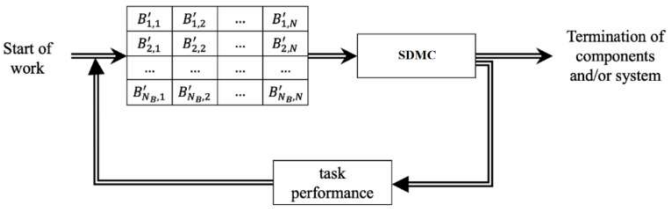

Figure 9. Refined structural diagram of the system S.

The sustainability degree of the system S during its functioning will be determined by the coefficient $k_{W^1_{S,c}}$ according to the value of the function $W^1_{S,c}$, calculated by formula (55), as follows:

$$k_{W^1_{S,c}} = \frac{\sqrt{\sum_{i=1}^{N_B}\sum_{j=1}^{N}\beta'^{\,2}_{i,j}}}{\sqrt{N_B \cdot N}}, \qquad (56)$$

Then, the system S at the value of $\frac{\sqrt{\sum_{i=1}^{N_B}\left(1-\alpha_z^{r,i}\right)^2}}{\sqrt{N_B}} \le k_{W^1_{S,c}} \le 1$, when the values of all $\alpha_z^{r,i}$ are the most permissible, will be in a stable state and the value of $k_{W^1_{S,c}}$ from this interval will be the sustainability criterion for this system. If some of the components of the system S are not active due to disabled computer stations, this will also affect the sustainability of its operation and, accordingly, the value of the coefficient will be lower, since it will take into account the need for all components of the system.

The next experiments were devoted to determine the value of the sustainability factor under different loads on the system S and under different system architectures in terms of the number of its components.

Let's set up and conduct the first experiment. The data obtained at a certain point in time of the system's operation were recorded under the following conditions: the system architecture was formed of all 100 components; subsystems responsible for detecting malware were not activated in the system in the absence of such manifestations. Under the conditions set at the beginning, the system should function stably. The results are presented in fifteen tables. For each characteristic indicator, its value for a particular component is given in the corresponding row and column. For this experiment, the significance levels of the characteristic indicators were set depending on their importance as follows:

1. $\alpha'_{1,S_i,1} = 0{,}01$, $\alpha'_{1,S_i,3} = 0{,}01$, $\alpha'_{2,S_{k+1,n},2} = 0{,}01$, $\alpha'_{2,S_{k+1,n},3} = 0{,}01$, $\alpha'_{2,S_{k+1,n},4} = 0{,}01$, $\alpha'_{2,S_{k+1,n},5} = 0{,}01$, $\alpha'_{3,S_{1,n},3} = 0{,}01$;
2. $\alpha'_{1,S_i,2} = 0{,}02$,$\alpha'_{3,S_{1,n},2} = 0{,}02$;
3. $\alpha'_{1,S_i,4} = 0{,}05$, $\alpha'_{1,S_i,5} = 0{,}05$, $\alpha'_{2,S_{k+1,n},1} = 0{,}05$, $\alpha'_{3,S_{1,n},1} = 0{,}05$, $\alpha'_{3,S_{1,n},4} = 0{,}05$, $\alpha'_{3,S_{1,n},5} = 0{,}05$.

Using formulas (55) and (56), we find the value of the function $W^1_{S,c}$ and the value of the sustainability coefficient of the

system $k_{W_{S,c}^1}$ and the value of the lower boundary of the interval for the sustainability coefficient $\frac{\sqrt{\sum_{i=1}^{15}\left(1-\alpha_z^{r,i}\right)^2}}{\sqrt{15}}$. The obtained values of $W_{S,c}^1 = 38.214301635550662$, $k_{W_{S,c}^1} = 0.98668902547623$ and the numerical value of the lower bound equal to 0.972848052541266 confirmed that the system is in a stable state.

For the second experiment, 30 computer stations were turned off. We got the same 15 tables, but each of them had at least thirty zero values. We calculated the values $W_{S,c}^1 = 31.96599695772904$, $k_{W_{S,c}^1} = 0.825358492414677$ and the numerical value of the lower bound, which was 0.813943077452799.

The third experiment was conducted with 40 computer stations turned off.

Let us perform an additional load on certain system components in order to influence their indicators.

We have calculated the values of $W_{S,c}^1 = 29.58811566557844$, $k_{W_{S,c}^1} = 0.763961861456294$ and the numerical value of the lower bound, which is 0.753564861176528.

In order to evaluate the malware detection accuracy using the proposed system S worm viruses were used. For this purpose five classes of worm-viruses were constructed. In addition, created worm-viruses had as malicious as benign functionality and were able to send to user message about computer station infection and its spreading in a computer network.

All computer stations used Windows operation system and all computer stations had the same configuration. The number of computer stations that have installed components of system S was 100, and 10 computer stations had no installed components of system S. The corporate network was divided into five segments, and it contained two servers.

In order to verify the worm virus detection accuracy several experiments with a partially distributed system. Experiment took into account six types of sources the worm virus may use for its spread. These sources were considered in the context of six possible infection options. During the experiments, the corporate network and an additional ten computer stations that do not belong to it were disconnected from the Internet.

Let's divide the corporate network into segments, with one of them designated as the demilitarized zone. For the experiments, a computer station outside of the demilitarized zone was randomly selected.

Each of computer stations contained a component of a partially centralized system, and each of stations were infected with artificial worm-virus.

For the experiment, we chose one of a set of 20 artificial worm viruses, with each of the five classes having four viruses. For this selected virus, we ran the full experiment in three series. Thus, in total, we conducted 360 series of experiments for all artificial worm viruses (12 series of experiments per worm virus class, multiplied by 5 classes, multiplied by 6 source variants).

The second variant of the experiment consisted of the fact that the computer station in which the artificial worm virus was activated was a computer station in the corporate network, but without the partially distributed system component installed.

The third scenario was similar to the second scenario, except that the computer station where the worm was activated was outside the corporate network. As a result, for these three scenarios, the one-to-all attitude was used, i.e., all computer stations on the corporate network were targeted for spreading the artificial worm virus.

In the following three variants, the "all to all" attitude was used, i.e., ten computer stations of the corporate network were targeted for the spread of the artificial worm virus.

In the fourth variant of the experiment, the same artificial worm virus was activated simultaneously in ten computer stations of the corporate network, each of which had components of the S system installed. In the fifth variant of the experiment, the same artificial worm-virus was activated simultaneously in ten computer stations of the corporate network, in which the components of the S system were not installed. The sixth variant of the experiment involved activating the same artificial worm-virus simultaneously in ten computer stations that did not belong to the corporate network.

When the same artificial worm virus was activated in one or ten computer stations, it could be that it could not gain control in all or some of them, and this event was also taken into account in the experiment design. The experiment duration for one of the five types of worm viruses of one series and one variant was 2 hours. Taking into account the known experience of the spread of real worm viruses, which was analyzed, the duration of their mass spread was relatively short in time, and the spread of more than one worm virus at the same time was not recorded. Therefore, we used only one copy of the artificial worm virus in each series of experiments, and the number of series for each of the twenty copies of artificial worm viruses was quantified as three. We analyzed and processed the results of the infection exclusively taking into account those computer stations that had the system components of the same artificial worm virus installed, i.e., one hundred computer stations.

A worm virus can infect a computer station only once, and it displays a message on the screen about the result of the infection. If it continues to infect files for a certain period of time, it will still report its presence once it has gained control.

Thus, during the experiments, the computer station could be infected or not infected due to the of the computer station configuration features, the presence of the S system, etc. It a necessary condition for ensuring the correctness of the experiment.

The computer stations include components of the system S for worm-viruses detection. The components can confirm the detection, which may coincide with the result of the message sent by the worm itself, or may diverge if the worm confirms the infection but the S system does not, or vice versa.

In the latter case, system S may correctly detect the worm.

Let us consider four variants of the events:

the type of worm-virus was detected and was assigned to one of the classes K_W^j $(j = 1,2,\dots,5)$;

1) a worm-virus was not detected by system S and infected the computer stations, but the corresponding artificial worm-virus informed about the successful infection of the computer station with its useful functionality, class $K_W^{0,j}$ $(j = 1,2,\dots,5)$, was filled;

2) a worm-virus was detected by system S, but the malware was assigned to a wrong class, user was not informed about the successful infection of the computer station, and then we will allocate it an additional class $K_W^{j,p}$ $(j = 1,2,\dots,5)$;

3) the computer station was not infected and the component and system S confirmed this and we will assign the class for this option as $K_W^{j,Y}$ $(j = 1,2,\dots,5)$.

The results of the experiments are presented in Table 1.

Let's analyze the results of the experiment. The proportion of True Positives Rate (TPR) is calculated as follows:

$$TPR = \frac{TP}{TP+FN} \cdot 100\% = \frac{10788}{14672} \cdot 100\% = 73{,}5278\%. \quad (57)$$

False Positives Rate:

$$FPR = \frac{FP}{TN+FP} \cdot 100\% = \frac{1935}{21328} \cdot 100\% = 9{,}0726\%. \quad (58)$$

To evaluate the accuracy of worm virus detection by the S system and the method implemented in it as an integral binary classifier, we will determine the sensitivity and specificity of the model and calculate their values. Sensitivity value:

$$S_e = TPR = 73{,}5278\%. \quad (59)$$

We define the specificity value as the proportion of true negative cases that were correctly identified and calculate it as follows:

$$S_p = \frac{TN}{TN+FP} \cdot 100\% = \frac{19393}{21328} \cdot 100\% = 90{,}9274. \quad (60)$$

Since the specificity value is high, the S system detects negative cases better than positive cases, because the sensitivity is lower than the specificity.

Table 1. Experimental results

Detection result	Classes worm-viruses, j= 1, 2, …, 5	Series of the experiment												Total
		Instances of the class												
		1			2			3			4			
		1	2	3	4	5	6	7	8	9	10	11	12	
FN	$K_W^{0,j}$	344	320	302	376	345	267	307	298	267	348	317	393	3884
TP	K_W^{j}	911	915	837	897	853	934	892	946	932	976	831	864	10788
FP	$K_W^{j,p}$	184	198	129	115	124	160	214	253	94	172	208	84	1935
TN	$K_W^{j,y}$	1561	1567	1732	1612	1678	1639	1587	1503	1707	1504	1644	1659	19393

VI. CONCLUSIONS AND FUTURE WORK

Security and protection of information in corporate networks is ensured by various means of different directions. Their uniqueness is very important in the context of active actions of intruders, who, due to lack of awareness of them, will have difficulties in carrying out malicious actions. The developed architecture of partially centralized systems makes it possible to create such tools that create problems for attackers in determining the center of their system and the principles of operation. The architecture model of such tools includes the possibility of dynamic configuration change, separation of the decision-making center, distribution of components by capabilities with the presence of a decision-making center in them and, therefore, it is the basis for further synthesis of adaptability and self-organization properties, the implementation of which will be carried out directly in the system components.

The developed architecture of the components of partially centralized systems is based on the obtained analytical expressions, which are mathematical models of the characteristic indicators of the values of the security levels of the components, formalizing the architecture of the system components S according to their functions, their purpose, interaction, place of execution, formation of the decision-making center and assessment of the security level of the computations performed.

A study of the sustainability of the developed distributed discrete system and experimental studies on the convergence of the values of the security levels of components depending on different system configurations, the number of components, and corporate network configurations are carried out. The results of the experiments con-firm the possibility of using the obtained mathematical models to form the architecture of system components.

The values of the characteristic indicators of the security levels of components of partially centralized systems will be the basis for forming decisions on its further steps and malware detection. Therefore, the directions of future work will be the detailing of partially centralized systems will be the use of values of characteristic indicators of the security levels of components in the organization of their functioning and in the identification of security threats.

References

1] Security information portal Virus Bulletin, threat landscape. [Online]. Available at: https://www.virusbulletin.com/ (accessed on 10.04.2023).

[2] The Independent IT-Security Institute. [Online]. Available at: https://www.av-test.org/en/ (accessed on 10.04.2023)

[3] Symantec Enterprise Cloud – Broadcom Inc. [Online]. Available at: https://www.broadcom.com/products/cybersecurity

[4] Symantec Product Categories. [Online]. Available at: https://sep.securitycloud.symantec.com/v2/landing

[5] SNORT. Foremost Open-Source Intrusion Prevention System. [Online]. Available at: https://www.snort.org/ (accessed on 12.04.2023)

[6] M. Van Steen, A. S. Tanenbaum, Distributed Systems, Third edition.; Preliminary version 3.01pre, 2017. ISBN: 978-90-815406-2-9.

[7] E. Tadmor, "Mathematical aspects of self-organized dynamics: Consensus, emergence of leaders, and social hydrodynamics," *SIAM News*, vol. 48, no. 9. 2015. [Online]. Available at: https://www.math.umd.edu/~tadmor/pub/flocking+consensus/SIAM%20News%2048(9)%207pp%20Tadmor%20self-organized%20dynamics.pdf.

[8] Y. Li, Y. Jiang, "Self-organization based service discovery approach considering intermediary utility," *Proceedings of the 2016 IEEE International Conference on Web Services (ICWS)*, 2016, pp. 308–315, https://doi.org/10.1109/ICWS.2016.47.

[9] F. Battiston, G. Cencetti, I. Iacopini, "Networks beyond pairwise interactions: Structure and dynamics," *Physics Reports*, vol. 874, pp. 1–92, 2020. https://doi.org/10.1016/j.physrep.2020.05.004.

[10] K. C. Laycraft, "Decision-making as a self-organizing process," *Ann. Cogn. Sci.*, Vol. 3, pp. 86–99, 2019. https://doi.org/10.1016/j.physrep.2020.05.004.

[11] B. T. Pentland, P. Liu, W. Kremser, T. Haerem, "The dynamics of drift in digitized processes," *MIS Quarterly*, vol. 44, pp. 19–47, 2020. https://doi.org/10.25300/MISQ/2020/14458.

[12] O. Kinouchi, R. Pazzini, M. Copelli, "Mechanisms of self-organized quasicriticality in neuronal network models," *Frontiers in Physiology*, vol. 8, article ID 583213, 2020. https://doi.org/10.3389/fphy.2020.583213.

[13] K. Katahira, Y. Chen, E. Akiyama, "Self-organized speculation game for the spontaneous emergence of financial stylized facts," *Physica A: Statistical Mechanics and its Applications*, vol. 582, article ID 126227, 2021. https://doi.org/10.1016/j.physa.2021.126227.

[14] N. Herakovič, H. Zupan, M. Pipan, J. Protner, M. Šimic, "Distributed manufacturing systems with digital agents," *Journal of Mechanical Engineering*, vol. 65, pp. 650–657, 2019. https://doi.org/10.5545/sv-jme.2019.6331.

[15] M. Neuer, "Cognitive perception and self-organization for digital twins in cyber-physical steel production systems," *Proceedings of the Industry 4.0 and Steelmaking Webinar of Steel Times International, Future Steel Forum, Prague*, Czech Republic, June 2020, https://www.researchgate.net/publication/342503882_Cognitive_percep

tion_and_self-organization_for_digital_twins_in_cyber-physical_steel_production_systems
[16] A. Darabseh, N. M. Freris, "A software defined architecture for cyberphysical systems," *Proceedings of the 2017 IEEE International Conference on Software Defined Systems (SDS)*, 2017, pp. 54–60, https://doi.org/10.1109/SDS.2017.7939141.
[17] A. Darabseh, N. M. Freris, "A software-defined architecture for control of IoT cyberphysical systems," *Cluster Computing*, vol 22, pp. 1107–1122, 2019. https://doi.org/10.1007/s10586-018-02889-8.
[18] K. Bellman, C. Landauer, N. Dutt, "Self-aware cyber-physical systems," *ACM Trans. Cyber-Phys. System*, vol. 4, 2020. https://doi.org/10.1145/3375716.
[19] L. Esterle, "Chapter 17 – Deep learning in multiagent systems," in Deep Learning for Robot Perception and Cognition, 2022, pp. 435-460, https://doi.org/10.1016/B978-0-32-385787-1.00022-1.
[20] N. Cointe, G. Bonnet, O. Boissier, "Ethics-based cooperation in multi-agent systems," *Advances in Social Simulation*, Springer, Cham, Manhattan, 2020. https://doi.org/10.1007/978-3-030-34127-5_10.
[21] K. Han, G. Kokot, O. Tovkach, A. Glatz, I. S. Aranson, A. Snezhko, "Emergence of self-organized multivortex states in flocks of active rollers," *Proceedings of the National Academy of Sciences*, vol. 117, pp. 9706–9711, 2020. https://doi.org/10.1073/pnas.2000061117.
[22] A. Pereira Junior, W. Pickering, R. Gudwin, Systems, Self-Organisation and Information, An Interdisciplinary Perspective, Routledge, Taylor & Francis Group, Oxfordshire, UK, 2018. [Online]. Available at: https://www.routledge.com/Systems-Self-Organisation-and-Information-An-Interdisciplinary-Perspective/Alfredo-Pickering-Gudwin/p/book/9781138609938.
[23] K. Wu, Q. Nan, "Information characteristics, processes, and mechanisms of self-organization evolution," *Complexity*, article ID 5603685, 2019. https://doi.org/10.1155/2019/5603685.
[24] Network Intrusion Detection System. [Online]. Available at: https://www.sciencedirect.com/topics/computer-science/network-based-intrusion-detection-system.
[25] What is a Wireless Intrusion Prevention System (WIPS)? Wi-Fi Security That's No Longer Up in the Air. [Online]. Available at: https://www.justfirewalls.com/what-is-a-wireless-intrusion-prevention-system/
[26] H. Ashtari, "What is network behavior analysis? Definition, importance, and best practices," *Network behavior analysis solutions collect and analyze enterprise network data to identify unusual activity and counter security threats*. [Online]. Available at: https://www.spiceworks.com/tech/networking/articles/network-behavior-analysis/.
[27] O. Pomorova, O. Savenko, S. Lysenko, A. Kryshchuk, "Multi-agent based approach for botnet detection in a corporate area network using fuzzy logic," *Communications in Computer and Information Science*, vol. 370, pp. 243-254, 2013. https://doi.org/10.1007/978-3-642-38865-1_16.
[28] O. Pomorova, O. Savenko, S. Lysenko, A. Kryshchuk, K. Bobrovnikova, "Anti-evasion technique for the botnet detection based on the passive DNS monitoring and active DNS probing," *Communications in Computer and Information Science*, vol. 608, pp. 83–95, 2016. https://doi.org/10.1007/978-3-319-39207-3_8.
[29] G. Suchacka, A. Cabri, S. Rovetta, F. Masulli, "Efficient on-the-fly Web bot detection," *Knowledge-Based Systems*, vol. 223, 107074, 2021. https://doi.org/10.1016/j.knosys.2021.107074.
[30] T. Sochor, M. Zuzcak, P. Bujok, "Analysis of attackers against windows emulating honeypots in various types of networks and regions," *Proceedings of the Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*, Vienna, Austria, 2016, pp. 863-868, https://doi.org/10.1109/ICUFN.2016.7537159.
[31] J. K. Murthy, "A functional decomposition of virus and worm programs," In: Qing, S., Gollmann, D., Zhou, J. (eds) Information and Communications Security. ICICS 2003. Lecture Notes in Computer Science, vol 2836. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-39927-8_37.
[32] Y. Desmedt, "Trojan horses, computer viruses, and worms," In: van Tilborg, H.C.A., Jajodia, S. (eds) Encyclopedia of Cryptography and Security. Springer, Boston, MA, 2011. https://doi.org/10.100.7/978-1-4419-5906-5_331
[33] A. Sheikh, "Trojans, backdoors, viruses, and worms," In: Certified Ethical Hacker (CEH) Preparation Guide. Apress, Berkeley, CA, 2021. https://doi.org/10.1007/978-1-4842-7258-9_5
[34] W. Shaojie, L. Qiming, "Analysis of a mathematical model for worm virus propagation," Advances in Information Security and Its Application. ISA 2009. Communications in Computer and Information Science, vol 36. Springer, Berlin, Heidelberg, 2009. https://doi.org/10.1007/978-3-642-02633-1_10.
[35] V. H. Pham, M. Dacier, G. Urvoy-Keller, T. En-Najjary, "The quest for multi-headed worms," In: Zamboni, D. (eds) Detection of Intrusions and Malware, and Vulnerability Assessment. DIMVA 2008. Lecture Notes in Computer Science, vol 5137, 2008. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-70542-0_13.
[36] F. T. Ngo, A. Agarwal, R. Govindu, C. MacDonald, "Malicious software threats," In: The Palgrave Handbook of International Cybercrime and Cyberdevianceб 2019 Palgrave Macmillan, Cham. https://doi.org/10.1007/978-3-319-90307-1_35-1.
[37] C. Edge, W. Barker, B. Hunter, G. Sullivan, "Malware Security: Combating Viruses, Worms, and Root Kits," In: Enterprise Mac Security, Apress, 2010. https://doi.org/10.1007/978-1-4302-2731-1_8.
[38] G. Connolly, A. Sachenko, G. Markowsky, "Distributed traceroute approach to geographically locating IP devices," *Proceedings of the Second IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, Lviv, Ukraine, 2003, pp. 128-131, https://doi.org/10.1109/IDAACS.2003.1249532.
[39] K. Bobrovnikova, S. Lysenko, B. Savenko, P. Gaj, O. Savenko, "Technique for IoT malware detection based on control flow graph analysis," *Radioelectronic and Computer Systems*, vol. 1, 2022, pp. 141-153. https://doi.org/10.32620/reks.2022.1.11.
[40] N. Lutsiv, T. Maksymyuk, M. Beshley, O. Lavriv, V. Andrushchak et al., "Deep semisupervised learning-based network anomaly detection in heterogeneous information systems," *Computers, Materials & Continua*, vol. 70, no. 1, pp. 413–431, 2022. https://doi.org/10.32604/cmc.2022.018773.
[41] V. Pevnev, V. Torianyk, V. Kharchenko, "Cyber security of wireless smart systems: channels of intrusions and radio frequency vulnerabilities," *Radioelectronic and Computer Systems*, no. 4, pp. 79-92, 2020.
[42] B. Savenko, S. Lysenko, K. Bobrovnikova, O. Savenko, G. Markowsky, "Detection DNS tunneling botnets," *Proceedings of the 2021 IEEE 11th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), IDAACS'2021*, Cracow, Poland, September 22-25, 2021, pp. 64-69. https://doi.org/10.1109/IDAACS53288.2021.9661022.
[43] N. Lutsiv, T. Maksymyuk, M. Beshley, A. Sachenko, L. Vokorokos, J. Gazda, "Deep semisupervised learning-based network anomaly detection in heterogeneous information systems," *Computers, Materials and Continua*, vol. 70, issue 1, pp. 413–431, 2021. https://doi.org/10.32604/cmc.2022.018773.
[44] O. Savenko, A. Sachenko, S. Lysenko, G. Markowsky, N. Vasylkiv, "Botnet detection approach based on the distributed systems," *International Journal of Computing*, vol. 19, issue 2, pp. 190-198, 2020. https://doi.org/10.47839/ijc.19.2.1761.

***SERGII LYSENKO**. Full Doctor, Full Professor of the Computer Engineering & Information Systems Department in Khmelnytskyi National University. The author has been working for more than 15 years in the field of increasing the effectiveness of detecting malicious software and cyber attacks, in particular: the theory and practice of creating multi-agent and distributed systems for detecting malicious software in computer networks, creating an adaptive technology for detecting cyber threats in computer networks.*

***Bohdan Savenko** PhD student of the Computer Engineering & Information Systems Department in Khmelnytskyi National University.
2021 – graduated from Khmelnytskyi National University with a degree in Computer Engineering (an educational and scientific program of the PR "Master"). Research interest: Methods and systems for detecting malicious software in computer networks*