PETYA IVANOVA PETROVA

MODELING OF NETWORK ATTACKS AND PROTECTION ALGORITHMS

AUTOREFERAT
FOR THE AWARD OF THE EDUCATIONAL AND SCIENTIFIC DEGREE " DOCTOR"
IN 4.6 INFORMATICS PROFESSIONAL DIRECTION

SCIENTIFIC LEADERS:

PROF. DSCTECH ANDON DIMITROV LAZAROV
VVMU “N.Y.VAPTSAROV”

PROF. DR. GEORGI GEORGIEV DIMITROV
UNIVERSITY OF LIBRARY SCIENCE AND INFORMATION TECHNOLOGY

BURGAS
2022

The dissertation paper was discussed and scheduled for defense by a decision of the Scientific Council of Burgas Free University - Burgas of April 15, 2022.

The doctoral student was enrolled in a part-time study form by Order LS-38 of December 07, 2015, with her name struck off with the right to defend by Order UMO 66 of January 31, 2020, of the BFU's Rector.

The dissertation consists of content, a list of abbreviations, a list of figures, an introduction, purpose and tasks, four chapters, used bibliography, thanks, and scientific publications to the dissertation paper. The written paper has a total volume of 114 pages and contains 7 figures and 2 tables. There are 2 appendices to the dissertation paper. Each chapter ends with conclusions. The bibliographic list includes 114 sources, of which 95 are in roman characters, 3 are in Cyrillic (Bulgarian) characters and 16 are Internet sources. The presentation is illustrated with 7 figures and 2 tables.
The designations of the formulas and graphs in the autoreferat correspond to those of the dissertation paper.
The defense of the dissertation paper will take place on June 24, 2022, at the Center for Informatics and Technical Sciences - Burgas Free University - Burgas before a scientific jury.

INTRODUCTION DISSERTATION PAPER'S PURPOSE AND TASKS

1. General characteristics of the processes in computer information systems

Cybersecurity incidents are daily occurrences. Examples are numerous, and their diversity spreads from the loss of information about individual users, attacks with malware and computer viruses, to large-scale criminal behavior provoked by organized crime. The predominant use of the botnet for the distribution of e-mail - spam, Distributed Denial of Service Attack, and the spread of malware has led to the activation of computer security experts worldwide. This is the reason for huge intellectual potential and computing resources to be directed to research and solve problems related to the cyber security of computer networks and information systems.

In this sense, building mathematical models of the operation of computer networks under malicious influences, and building highly effective methods and algorithms for predicting and countering cyberattacks, are essential for solving problems of ensuring sustainable cybersecurity in computer networks.

Modeling the behavior of computer networks includes defining the space of invariant parameters of the behavior of computer networks, which is an approach that allows adequate modeling of the complex structure and variety of the Internet components.

Telecommunication traffic is being modeled as a statistical process with the probabilistic distribution of Poisson requests and responses. In fact, research on Internet data traffic shows that Poisson's simple models do not adequately reflect the behavior of the real network traffic, including local and global network traffic. The behavior of the local network can be modeled as a self-similar process, point process, marked point process, and time series.

5. **Dissertation paper's purpose and tasks**

The dissertation paper's purpose can be defined based on the analysis of processes related to the cybersecurity of computer networks.

Mathematical modeling of processes in computer networks when impacted by malware, construction and application of genetic algorithms to detect intrusions into the computer network and data protection.

The following basic tasks can be defined in accordance with the purpose formulated:

- Modeling processes when a computer network is impacted by malware.
- Construction and application of genetic algorithms to detect intrusion into the computer network.
- Determining the Fitness function in the genetic algorithm to detect intrusion into computer networks.
- Implementation of network security through encrypting with a genetic algorithm.

CHAPTER I.
MODELING THE PROCESSES WHEN COMPUTER NETWORK IS IMPACTED BY MALWARE

The subject of this chapter are mathematical analysis and the model of the processes of susceptibility, exposure, infection and restoration of computer networks when impacted by malware. The behavior of the network is described with a system of differential equations. Two cases are analyzed: case of equilibrium in the computer network and case of imbalance in the computer network. Analytical expressions have been obtained to calculate network characteristics in case of susceptibility, exposure, infection and recovery of computer nodes during a malware attack.

1.1. Introduction to the problem and formulation of the task

The task of this chapter is to build a mathematical model of the computer network's behavior when predisposed (susceptible), exposed, infected and recovered after an attack with malicious

software, i.e. determining the number of units in the computer network prone to attack, exposed to impact, infected and recovered after impact.

Mathematical model of cyber attack: Solution of a system of differential equations in a computer network equilibrium

Basic values and parameters:
S - the number of those susceptible to the impact of malware;

E - the number of exposed to malware infection, i.e. the number of infected computers that do not infect other computers,
I - the number of computers infected with malware, i.e. the number of infected computers that infect other computers,

R – number of computers recovered,
k calculated transfer capacity, $k = S + E + I + R$
β is the degree of infectious contact,
δ is the degree of failure of nodes in the network as a result of infection,
τ is the degree of infection of the exposed unprotected class exposed to infection,
μ is the mortality rate due to a malware attack.

r - the natural rate of increase in the number of Class S computers.

- After malware attack recovery function

$$\mathrm{Rec}(I) = \begin{cases} \rho I, & 0 \le I \le I_{\min} \\ m, & I \ge I_{\min} \end{cases} \quad (1)$$

- $I_{\min}$ is the minimum number of infected nodes, then the antivirus program is turned on, $m = \rho . I_{\min}$

- The dynamics of the nodes of the susceptible *S* class is defined by the rate of change of S

$$\frac{dS}{dt} = r.S - (r.S).\left(\frac{S}{k}\right) - \frac{\beta.I}{k}.S - \delta.S \quad . \tag{2}$$

- The dynamics of the nodes of the exposed E class is defined by the rate of change of E

$$\frac{dE}{dt} = \frac{\beta.I}{k}..S - (\tau + \delta).E \tag{3}$$

- The dynamics of the nodes of the infected I class is defined by the rate of change of E

$$\frac{dI}{dt} = \tau.E - (\mu + \delta).I - \mathrm{Re\,c}(I) \tag{4}$$

- The recovery dynamics of R-class nodes is defined by the rate of change of R

$$\frac{dR}{dt} = \mathrm{Re\,c}(I) - \delta R \tag{5}$$

Solution of a system of differential equations in a computer network equilibrium, i.e.

$$\frac{dS}{dt} = 0 \quad , \quad \frac{dE}{dt} = 0 \quad , \quad \frac{dI}{dt} = 0$$

- when $I < I$ min

$$S' = \frac{a(\delta + \tau)}{\beta.\tau} \quad E' = \frac{a.r.(k\beta - a(\delta + \tau)) - k.\delta.\beta.\tau}{\beta^2.\tau^2.k} \quad I' = \frac{.r.(k\beta - a(\delta + \tau)) - k.\delta.\beta.\tau}{\beta^2.\tau.k} \tag{11}$$

where $a = \rho + \mu + \delta$

- when $I > I$ min

$$I_{1,2} = \frac{n \pm \sqrt{t}}{2.\tau.k.\beta^2} \quad , \tag{16}$$

$$S_{1,2} = \frac{2.\tau.k.\beta(r - \delta) - n \pm \sqrt{t}}{2.\tau.r.\beta} \tag{17}$$

$$E_{1,2} = \frac{(\mu + \delta)(n \pm \sqrt{t}) + 2.m.\tau.k.\beta^2}{2.k.\tau^2..\beta^2} \tag{18}$$

where $n = \beta\tau k(r - \delta) - r(\tau + \delta)(\mu + \delta)$, $t = n^2 + 4[\beta^2\tau.k.m..r.(\tau + \delta)]$.

- The endemic equilibrium condition when an antivirus program is enabled

$$\sqrt{t} \geq I_{\min}.2.\tau.k.\beta^2 - n \tag{20}$$

Solution of a system of differential equations in a computer system imbalance condition

$\frac{dS}{dt} \neq 0 \; \frac{dE}{dt} \neq 0 \; \frac{dI}{dt} \neq 0$

- The system of differential equations for *S, E, I, R* is being written in the form

$$\begin{aligned} \frac{dS}{dt} &= r.S - (r.S).\left(\frac{S}{k}\right) - (\beta.I).S - \delta.S \\ \frac{dE}{dt} &= (\beta.I).S - (\tau + \delta).E \\ \frac{dI}{dt} &= \tau.E - (\mu + \delta).I - \rho.I \\ \frac{dR}{dt} &= \mathrm{Re\,c}(I) - \delta R \end{aligned} \tag{40}$$

The solution of the system of differential equations has the form:

- For the computer nodes class, prone to be attacked

$$S(t) = \frac{E_0.e^{-a.t}}{1 + E_0.(b/a).e^{a.t}} \tag{51}$$

where $a = (r - \beta.I_{\min} - \delta)$, $b = \left(-\frac{r}{k}\right)$.

- For the computer nodes class, exposed to attack

$$E = E_0.e^{-\omega.t}, \tag{70}$$

where

$$\omega_{1,2} = \frac{-(\tau + 2\delta + \mu + \rho) \pm \sqrt{(\tau + 2\delta + \mu + \rho)^2 - 4\left[(\tau + \delta)(\mu + \delta + \rho) - \frac{\tau.\beta.E_0.e^{-a.t}}{1 + E_0.(b/a).e^{a.t}}\right]}}{2} \tag{66}$$

where $\omega > 0$.

- For the infected computer nodes class

$I = I_0 . e^{-\omega . t}$

- For the restored computer nodes class

$R(t) = m\left(1 - e^{-\delta . t}\right)$,

where $m = \rho . I_{min}$.

1.6. Numerical experiment to evaluate the impact of malware on a computer network

The numerical experiment is to be performed at the following initial values of the main parameters vulnerability $S_0 = 93$, exposure $E_0 = 5$, infection $I_0 = 2$, recovery $R_0 = 0$, rate (degree) of contact infection $\beta = 0.05$, rate (degree) of destruction (failure) of nodes $\delta = 0.02$, rate of infection $\tau = 0.04$, rate of intrusion $r = 0.2$, calculated transfer capacity with increasing $k = 100$, rate of recovery $\rho_0 = 0.03$, rate of destruction as a result of attack $\mu = 0.01$.

The symbol b indicates the degree of inclusion of new nodes in the susceptible class, μ is the degree of mortality due to an attack with malware (virus), β is the degree of infectious contact, δ is the degree of failure of nodes in the network as a result of infection, τ is the degree of infection of the exposed unprotected class exposed to infection.

Visualization of the results of the experiment

Infection, when endemic equilibrium, increases with the time of exposure, which is initially steep, after which the rate of increase decreases (Fig. 1). This course of the infection curve $I_1(t)$ follows the law of change of susceptibility (vulnerability) $S_1(t)$ (Fig. 2) and the exposure (exposition) $E_1(t)$, and (Fig. 3) change of the network due to malware.

Dependence of susceptibility (vulnerability) on the time, when endemic imbalance, is shown in Fig.1.

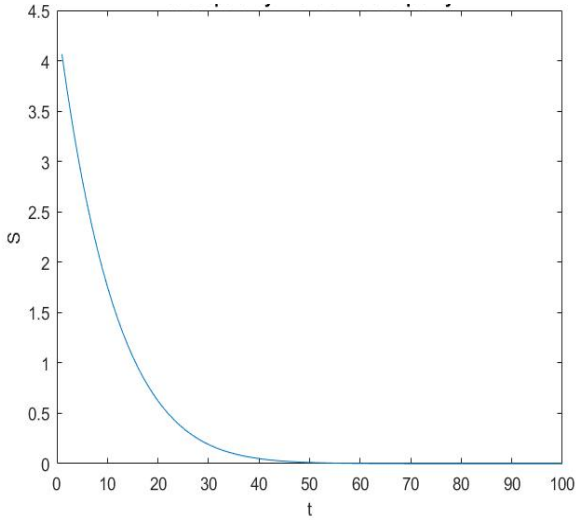

Fig. 1

Fig. 1. Dependence of susceptibility (vulnerability) on the time, when endemic imbalance.

Dependence of recoverability on the time, when endemic imbalance, i.e. when security software is operating, is shown in Fig. 2.

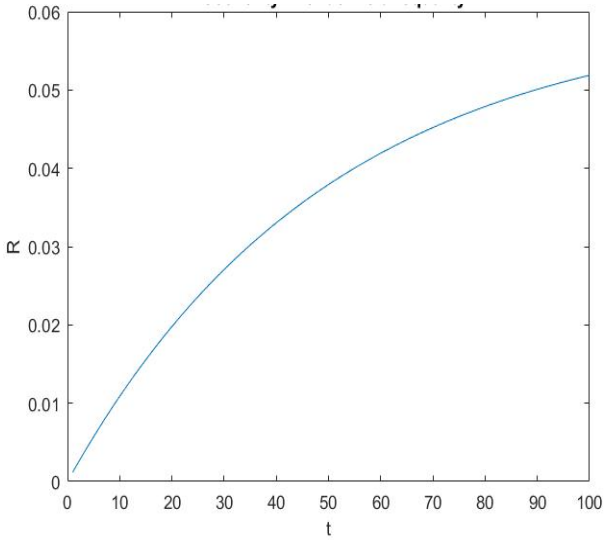

Fig. 2

Fig. 2 Dependence of recoverability on the time, when endemic imbalance, i.e. when security software is operating.

Dependence of recoverability on the time, when endemic imbalance, i.e. when security software is operating, is shown in Fig. 3.

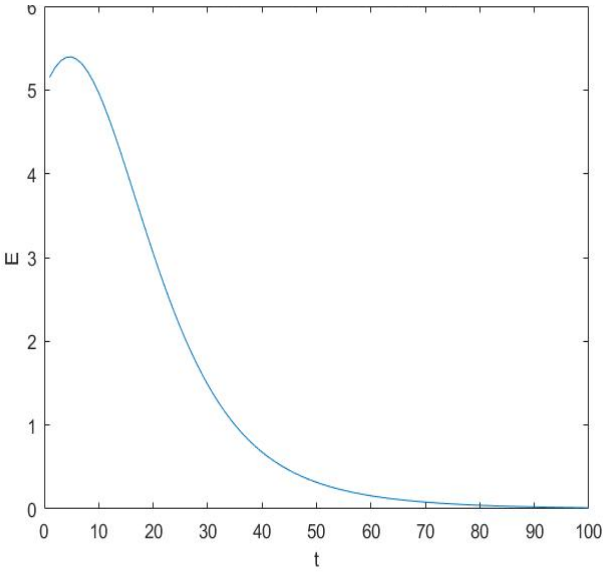

Fig. 3

Fig. 3. Dependence of exposure on the time, when endemic imbalance, i.e. when security software is operating.

Dependence of infection on the time, when endemic imbalance, i.e. when security software is operating, is shown in Fig. 4.

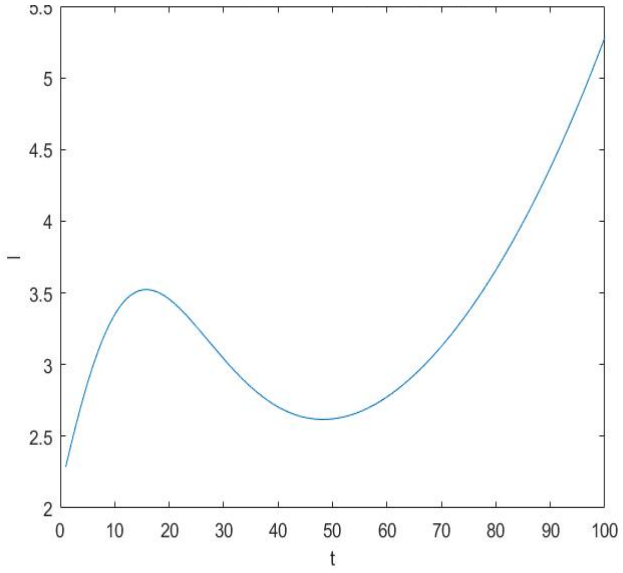

Fig. 4

Fig. 4. Dependence of infection on the time, when endemic imbalance, i.e. when security software is operating.

The analysis of the results shows that in endemic equilibrium, i.e. in the absence of security software, the indicators characterizing the behavior of the computer network increase with the change of time, which at the beginning of the curve is at high speed.

When endemic imbalance, i.e. security software is available, the parameters have a complex dependence on the time. Of interest are the dependences of exposure and infection on time. At the beginning of the observation interval, there is a short-term exponential increase in exposure when the security software is not in the active phase (t = 5). The infection increases until t = 15, then decreases until t = 60, when the time for software update occurs. The increase in infection is due to the non-updated security functions of the security software.

1.7. Conclusions

The basic differential equations describing the state of the computer network under the impact of malware are given. Solutions

of the differential equations in an equilibrium of the computer system and in the absence of equilibrium in the processes of susceptibility, exposure, infection and recovery after an attack with malware have been obtained. Original solutions of the inhomogeneous differential equations and the system of inhomogeneous differential equations are applied.

The developed methodology for estimating the behavior of the computer network can be applied to an a priori known law on influence, subject, as a rule, to the Poisson distribution.

CHAPTER II
CONSTRUCTION OF GENETIC ALGORITHMS TO DETECT INTRUSION INTO THE COMPUTER NETWORK

The subject of this chapter is the analysis of the main functions and structure of the genetic algorithm for detecting intrusions into computer networks. The sequence of characteristics of network communications is interpreted as a chromosome, which determines a rule for detecting intrusions, while the characteristics themselves are considered as genes of the chromosome. The main network characteristics of IP v-4 and IP-v6 and their structure are provided and illustrated in examples of intruded and non-intruded network connections.

The task of this chapter is to build the syntax of the genetic algorithm and analyze the network structure and characteristics (genes and their codes) of the rules (chromosomes) to determine the type of communication exchange.

In accordance with the task thus defined, a general characteristic of a system for detecting intrusions into computer networks is being made. The types of intrusions into computer networks are being considered. When the characteristic is made, then the parameters of the genetic algorithm used to detect and detect attacks in computer networks are defined. The syntax and structure of different rules (chromosomes) in the genetic algorithm are defined. The process of coding network characteristics (genes) is described in detail and illustrated with examples.

2.5. Defining the syntax and structure of the rule (chromosome) in the genetic algorithm

Genetic algorithms are used to generate network characteristics (rules) for estimating computer network traffic. These rules are used to determine and differentiate between normal and anomalous network connections. Аномалните връзки са тези, които се отнасят до събития, които е много вероятно да бъдат оценени като злонамерени прониквания. The rules stored in the rules (knowledge) database of the intrusion detection system have the syntax of a predicate logic clause [86, 94]:

If {condition} then {act}

The part defines a state (condition) described by network characteristics, such as source IP addresses and destination and port numbers (TCP/IP network protocols used), communication duration, etc., including an indication of the probability of intrusion. This part of the rule compares the genetic algorithm with the network characteristics of the rules stored in the rules database of the intrusion detection system. The characteristics in the conditional part are connected by a logical AND operator. The *act* section refers to an action defined by the security rules, such as a warning report to the system administrator, stopping communication, logging a message in system monitored files, or all of the above. Some network features have a greater relative contribution to the definition of network connections and communication exchange.

2.6. Coding of network characteristics (genes) of rules (chromosomes)

Chromosome genes can be represented by different types of data, binary numbers (bytes), decimal numbers, or floating-point numbers. This is determined by the different formats and range of data values for the different network characteristics. Table 1 shows the name of the network characteristics - chromosome attributes, the number of genes defining the attributes of chromosomes, and their formats, respectively in the first, second and third columns.

For example, the "Duration" characteristic has three components (hours, minutes and seconds), each of which is represented by one byte-type gene (Table 1). Similarly, each of the characteristics "Protocol", "Source Port Number", "Destination port number" is encoded using one gene of integer type, and each of the characteristics "Source IP-v4" and "Destination IP-v4" have four components. (a, b, c and d), each represented by a single byte gene, "Source IP-v6" and "Destination IP-v6" have eight components. (a, b, c, d, e, f, g, h), each of which is represented by one byte-type gene.

Name of chromosome attributes	Number of genes	Code format
Source IP-v 4 address	4	a.b.c.d
Source IP-v 6 address	8	a.b.c.d.e.f.g.h
Destination IP -v 4 address	4	a.b.c.d
Destination IP -v 6 address	8	a.b.c.d.e.f.g.h
Source Port Number	1	Integer
Destination Port Number	1	Integer
Duration	3	h:m:s
State	1	Integer
Protocol	1	Integer
Number of Bytes sent by Originator	1	Integer
Number of Bytes sent by Responder	1	Integer
Attack_name	1	Integer

Table 1 Attributes, genetic structure and codes of the network characteristics of the chromosome (rule) in the genetic algorithm.

The “Attack name” attribute is located in the *act* part of the rule, which classifies the network characteristics at the learning stage or determines the nature of the communication at the intrusion detection stage when the *condition* of a particular rule coincides with that of the training stage.

An example of a rule that classifies network communication as a *Neptune* Denial of Service (DoS) attack is the following chromosome [30]:

if (duration=“0:0:1” and protocol=“finger” and source_port=18982 and destination_port=79 and source_ip=“9.9.9.9” and destination_ip=“172.16.112.50”) then (attack_name=“neptune”).

The rule states that if the network packet originates from IP address 9.9.9.9 and port 18982 and is sent to IP address 172.16.112.50 and port 79 using the finger protocol, and the communication duration is 1 second, then it is most likely to be a Neptune network attack that can shut down the destination host (packet destination). Each rule is to be encoded as a chromosome using a fixed-length vector, where each network characteristic is to be encoded using one or more genes of different types (second and third columns of Table 1). The coded chromosome of the rule in the example above takes the form:

{0 0 1 2 18982 79 9 9 9 9,172 16,112 50 1}

Example of a rule that outputs a stop communication action:

if {the connection has following information: source IP address 124.12.5.18; destination IP address: 130.18.206.55; destination port number: 21; connection time: 10.1 seconds} then {stop the connection}

This rule can be interpreted as follows: if there is a request for network communication with source IP address 124.12.5.18, destination IP address (destination) 130.18.206.55, destination target port number 21, and communication time 10.1 seconds, then the establishment of this communication connection is stopped.

In other words, the 124.12.5.18 IP address is recognized by the intrusion identification system as one of the IP addresses in the

blacklist; therefore, any request for a service function, initiated by it, is being rejected.

The chromosome attributes are as follows: Source IP address, Destination IP address, Source port number, Destination port number (destination), Duration (duration of communication in seconds), Communication status, Protocol, Number of bytes sent by the sender, Number of bytes sent by the receiving correspondent.

An example of a rule with chromosome attribute values:

if {the connection has following information: source IP address 209.11.??.??; destination IP address: 130.18.176+?.??; source port number: 42335; destination port number: 80; connection time: 482 seconds; the connection is stopped by the originator; the protocol used is TCP; the originator sent 7320 bytes of data; and the responder sent 38891 bytes of data} then {stop the connection}

In this case, the encoded chromosome has the form [11]:

/d 1 0 b -1 -1 -1 -1 8 2 1 2 b -1 -1 -1 4 2 3 3 5 0 0 0 8 0 0 0 0 0 0 0 4 8 2 1 1 9 0 0 0 0 0 0 7 3 2 0 0 0 0 0 0 3 8 8 9 1/

The rule can be interpreted as follows: if network communication with the source IP address 209.11. ??. ?? (209.11.0.0 ~ 209.11.255.255), Destination IP address 130.18.176. ?? (130.18.176.0 ~ 130.18.255.255), source port number 42335, destination port number 80, duration 482 seconds, ending with state 11 (communication terminated by creator), using protocol type 9 (TCP), and source sends 7320 bytes of data, respondents send 38891 bytes of data, then this is suspicious behavior and can be identified as a potential intrusion.

The validity of this rule is assessed by matching a preliminary set of data composed of connections (communications) marked as anomalous or normal. If the rule is able to find anomalous behavior, a "reward" will be given to the current chromosome. Obviously, no rule can be used to segregate all anomalous connections from normal

connections. Obviously, no rule can be used to segregate all anomalous connections from normal connections. The population is evolving to find the optimal set of rules.

The symbols ("*" and "?") are used to denote wildcards (network parts of IP addresses), with the corresponding genes on the chromosome represented by -1. These wildcards are used to represent a range of specific network addresses, i.e. presentation of a network block (range of IP addresses or port numbers) in a rule. Once information about the characteristic field is included in the rules, the ability of the intrusion detection system can be improved, as intrusion can start from many different addresses [86]. Including the duration of network communication in the chromosome ensures the inclusion of time information about network connections. The maximum value of the duration is 99999999 seconds, which is more than one year. This is necessary to identify complex intrusions, which can take hours, days or even months. It will be emphasized once again that the genetic algorithm starts with an initial population, which consists of randomly selected rules and is developed using the operators of crossover and mutations. In accordance with the effectiveness of the suitability assessment function, subsequent populations are subject to rules that correspond to intruding connections. At the end of the genetic process, the algorithm stops, and the rules are selected and added to the basis of the system for detecting intruding connections.

The rules for classifying DoS attacks (Smurf, Mailbomb), R2L Warezmaster, multihop), U2R (Snmpguess, Buffer-overflow), Probing (ip-sweep, saint), derived from the DARPA-USA Intrusion Database, have the following structure [91]:
DOS:
Rule 1 – *if duration = 0 and protocol_type = tcmp and dst_host_srv_count = 255 and then Smurf*
Rule 2 – *if duration = 1 V 5 V 11 and protocol_type-tcp and dst_host_srv_count >= 2^ <= 247 and then Mailbomb*
R2L:
Rule 3 – *if duration = 0v duration <=289 and protocol_type = tcp and dst_host_srv_count >=1^ <= 128 and then waremaster*

Rule 4 – *if duration = 0 and protocol_type = icmp V top V udp and dst_host_srv_count >= 1^ <= 20 and then multihop*

U2R:
Rule 5 – *if duration = 0 V duration <= 289 and protocol_type = udp and src_bytes-I > and then Snmpguess*
Rule 6 – *if 1 and = 0 and protocol_type = tcp and dst_host_srv_count <= 100 and then buffer–overflow*

Probe:
Rule 7: *if duration = 0 and protocol_type – icmp and dst_host_srv_count >= 1^ <=255 and then ipsweep*
Rule 8: if duration = 0 and duration <= 11 and protocol_type – icmp V tcp V udp and dst_host_srv_count >=1^ <=255 and then saint

The definition of the above rules uses the following notations "<=" (less than or equal to), "=>" (equal or greater), ^ (logical "and").

2.7. Basic operations on chromosomes of a given generation and experimental evaluation of the fitness function of Firas Alabsi

Selection

When generating each succeeding generation, part of the resulting population is chosen to create a new generation. Decisions to select individuals (chromosomes or rules) are derived through a based process, which ensures that individuals with higher values of fitness function are selected with a high probability. Pairs of chromosomes are selected from the population, to be the parents of the next population, i.e. to recombine chromosome pairs.

Crossover (recombination)

Crossover or recombination creates one or more new generations of parent chromosomes to produce better chromosomes with high suitability function values.

Mutation

Mutation randomly changes the new progeny. This is done to prevent all decisions in the chromosome population from falling into the local optimum when making a decision.

Fig. 5 shows an experimental estimate of the evolution of current chromosome generation by fitness function (FF) and test chromosome, recombination and mutation, and new chromosome generation in binary format; recombination chromosomes are indicated by indication ♀ and ♂; an invalid chromosome is denoted by ●, and unchanged chromosome is denoted by ○ , a mutated chromosome is denoted by ○ .

Current Generation	FF	Crossover&Mutation	New Generation
1011101010001010	0.63♀		1011101010001010
1000011110101000	0.43●	1011100100001001	1011100100001001
1100101000101011	0.50●	0101111010001010	0101111010001010
0101110100001001	0.75♂		0101110100001001
0101000111010101	0.56◌	0101000011010101	0101000011010101
0010111000101011	0.56○		0010111000101011

↑

Test Chromosome
0101111010001101

Fig. 5.

Fig. 5. Evaluation of current chromosome generation and Test Chromosome, by recombination and mutation fitness function, and new chromosome generation in binary format; ♀ and ♂ recombination chromosomes; ● invalid chromosome, ○ a chromosome without change, ◌ mutated chromosome.

The current chromosome and test chromosome function is calculated as the ratio of the number of matching bits in the chromosomes to the number of bits in the test or comparable chromosome, i.e.

$$Fitness = M_b/N_b,$$

where M_b is the number of matching bits in the test and comparable chromosomes, N_b is the number of bits in the chromosome - test and current comparable.

In this case, the total number of bits in the chromosome is 16.

2.9. Conclusions

The idea of applying genetic algorithms in systems for detection and prevention of intrusions into computer networks has been developed. For the assessment of current and new intrusions, the use of integrated and complementary assessment fitness functions is envisaged, which will increase the accuracy of the assessments and the prevention effectiveness concerning intrusions in computer networks.

The fitness function for estimating network characteristics (genes) in the detection of intrusion into computer networks based on a genetic algorithm is applied. A generalized genetic algorithm and a pseudo-code are adapted. An experimental evaluation of the Firas Alabsi method for calculating the fitness function of the generated rules (chromosomes) has been performed.

As a future activity of the author in the field of detecting malicious intrusions into computer networks by applying the approach of genetic algorithms, it is planned to build a database (rules, knowledge) for new unexplored structures of network characteristics of computer attacks and methods for their counteraction and prevention.

CHAPTER III
IMPLEMENTATION OF NETWORK SECURITY USING AN ENCRYPTING GENETIC ALGORITHM

3.1. General characteristics of the combined encryption algorithm Security in data transmission is a major problem in communication systems.

Security in data transmission is a major problem in communication systems. The subject matter of this chapter is the security of the transmission of confidential information and data with the development of a public algorithm for cryptography of plain text using a genetic algorithm to ensure confidentiality, authenticity, integrity, and trouble-free transmission of information. The protection of information is being achieved by applying various encryption techniques. One of the powerful encryption methods is the asymmetric RSA (Rivest-Shamir-Adleman) encryption, which can be further enhanced by the use of artificial intelligence tools such as the genetic algorithm. Unlike the RSA encryption algorithm, which starts the selection of two prime numbers, the dissertation paper develops an encryption algorithm based on a random selection of two mutually non-prime numbers, which is amplified by using basic operations of the genetic algorithm - recombination and mutation. The mutation is achieved by concatenation of the two chromosomes, one of which has an inverse transcript of the cipher symbols.

The scheme of encryption of the message in the transmitter is presented in (fig. 6). In the separate block the public and the private key are generated on the basis of two unequal numbers, randomly generated in the interval 10 - 1000. The second block presents the input data, i.e. the transmitted message to be encrypted. The encryption in the third block is being performed by a public key operation above the input data.

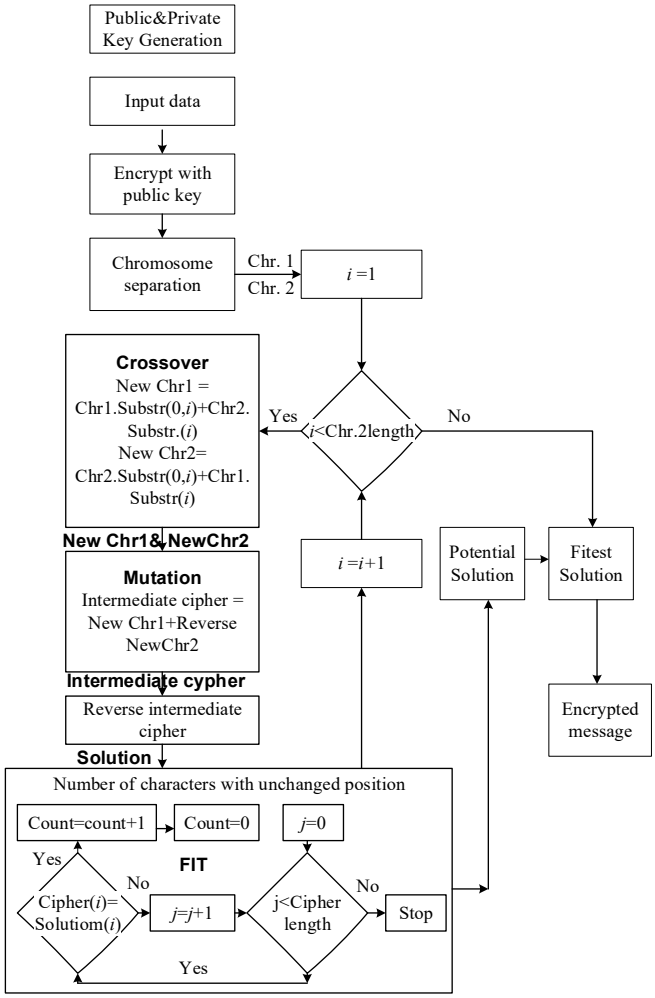

Fig. 6 Block diagram of message encryption and increasing the stability of the cipher through a genetic algorithm

The string of the encrypted message is divided into two equal parts of length n/2. The first part is the first chromosome, the second part is the second chromosome. A cycle with an index in the range of 0 to the length of the second chromosome n/2 is being created. The following operators apply to each i-step of the cycle:

Description of the algorithm

First, a public and a private key are being generated by applying an algorithm using two not mutually prime numbers. Two prime numbers are being used in RSA encryption. A text message is being submitted, which is encrypted through a public key. An encrypted message (cipher) is being generated. The encrypted text is being divided into chromosomes.

The string of the encrypted message is being divided into two equal parts of length $n/2$. The first part is the first chromosome, the second part is the second chromosome. A cycle with an index in the range of 0 to the length of the second chromosome $n/2$ is being created. At each i-th step of the cycle the following operations of the genetic algorithm are to be applied:

Crossover

Two new chromosomes are generated from the two chromosomes by concatenation of two separate parts of the two chromosomes, determined by the i-th point of recombination (crossover):

New Chromosome 1 = Chromosome 1 Substring (0, i) + Chromosome 2 Substring (i)

From the first chromosome, a substring of symbols, of a length 0 to i, is taken. From the second chromosome, a substring of symbols, of a length i, is taken.

New Chromosome 2 = Chromosome 2 Substring (0, i) + Chromosome 1 Substring (i)

From the second chromosome, a substring of symbols, of a length 0 to i, is taken. From the first chromosome, a substring of symbols, of a length i, is taken.

where i is the point of recombination or crossover.

Mutation

The mutation is being realized by concatenation of New Chromosome 1 with Reverse New Chromosome 2.

An intermediate cipher (intermediate encrypted message) is being obtained, which after inversion is a potential solution for an encrypted message.

The mutation operator is being applied to shift the positions of the symbols in the encrypted message without changing the symbols themselves, which increases the entropy of the cipher. The mutation is performed by concatenation of New chromosome1 with the inverse New chromosome 2.
The resulting cipher string manipulated by the mutation operator generates an intermediate cipher, i.e. the intermediate encrypted message. The intermediate encrypted message is to be inverted. Cipher inversion further shifts the positions of the characters and increases the cipher's entropy. Intermediate cipher inversion is a potential solution for an encrypted message.

Cipher evaluation with fitness function, operations in FIT block

Whether the potential solution for the encrypted message is a solution for the final cipher is evaluated by determining how many characters in the encrypted message remain with unchanged positions in the intermediate cipher through a special fitness function. The public key encrypted message is to be compared with the potential solution of an encrypted message obtained after applying the operations of the genetic algorithm recombination and mutation. The minimum value of the fitness function is a criterion for solving the cipher.
A cycle is formed to evaluate the position of each j-th character of the encrypted message, establishing whether it has changed its position in relation to the encrypted public message. The counting operator determines how many characters from the encrypted message and the one manipulated by the genetic algorithm remain unchanged.
The potential solution of the cipher, the string with a minimum number of characters with unchanged positions, i.e. the string with the minimum fitness function is the cipher solution.
In the receiver, i.e. at the decryption stage, the operators are to be applied in reverse order (Fig. 7).

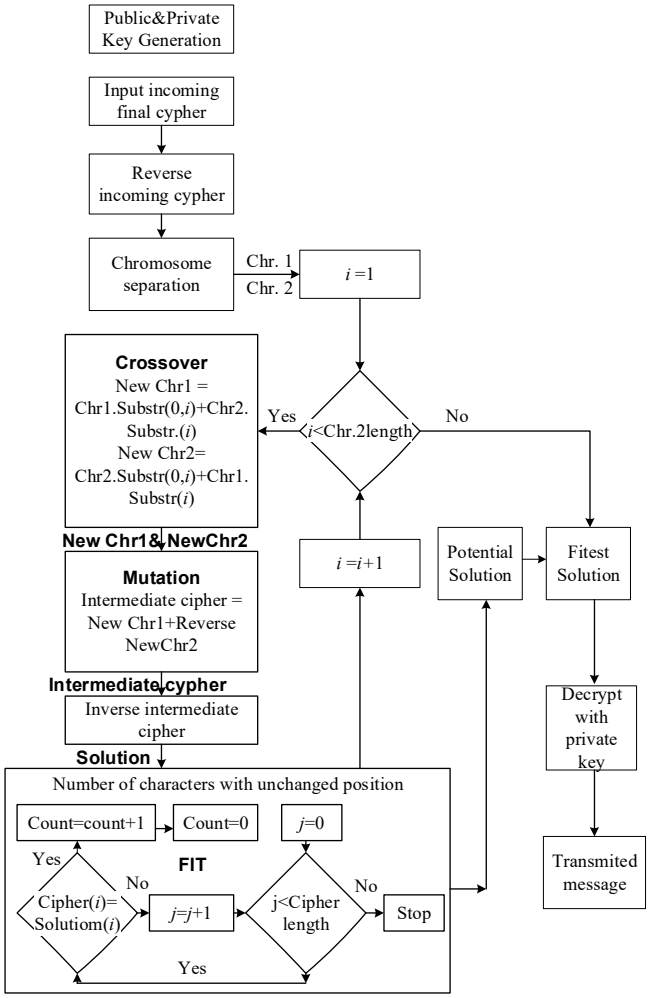

Fig. 7. Block diagram on decrypting the cipher and receiving the transmitted message

First, the operators of the genetic algorithm are being used to recover the cipher message. The private key is then being used to recover the message. The scheme of decryption of the cipher in the receiver is presented in (fig. 7). The public and private keys are presented in the first separate block. The second block presents the input data, i.e. the

encrypted message to be decrypted. The first operation above the cipher's string of numbers is the inversion of the cipher string. The cipher sequence is then divided into two chromosome sequences of equal length and the procedure continues as in the encryption phase, including operators, chromosome crossover, mutation by concatenating the first chromosomes with the inverse version of the second chromosome, cipher inversion and definition of potential solution through fitness function. The final cipher with the minimum fitness function is decrypted with a private key to receive the transmitted message.

3.2. Structure of an encryption algorithm, reinforced with the operators of the genetic algorithm

The structure of the encryption algorithm and the implementation methods can be defined as follows.

- To generate a key pair /public and private/. In asymmetric-key cryptography, the rule is that when a text is encrypted through a public key, it is decrypted with the corresponding private key and vice versa. Plain text is encrypted through a public key to produce an intermediate cipher. The intermediate cipher is encrypted through a genetic algorithm to produce the last cipher.

- Decryption method:

The final cipher is decrypted using a genetic algorithm to obtain an intermediate cipher, which is re-decrypted using the appropriate private key to obtain plain text.

- **Initial encryption:**
  Intermediate cipher = encryption /plain text, public key/
  OR
  Intermediate cipher = encryption /plain text, private key/
  AND
  Genetic encryption: final cipher = encryption / intermediate encryption /
- **Genetic decryption:**
  Intermediate encryption = decryption / final cipher /

Final decryptor: plain text = decryption / intermediate cipher, private key /

OR

Plain text = decryption / intermediate cipher, public key /

Basic steps and components of the genetic algorithm

1. Keys generating :

Step 1: Generate two non-coprime number /a, b/
Step 2: Set 1: = L.C.M /a, b/ g: = G.C.D /a, b/
Step 3: Set /a, g/ as private key
Step 4: Set x: = ((a-1)/g) + ((a-1)%g) y: = g
Step 5: Store x, y
Step 6: Set (b, l) as public key
Step 7: Set p: = ((l-1)/b+((l-1)%b) q: = b
Step 8: Set p, q
Step 9: Stop
Function:
Input: text, key type, public key (p, q) or private key (x, y)
Output: Intermediate Cipher
Step 1: if key type = private key Read public key (p, q) from database Set m: = 2 else Read private key (x, y) from database Set m: = 1
Step 2: Set x: = ((x-(y-1))*y)+y
Step 3: Set p: = ((p-(q-1))*q)+q
Step 4: if (x/y = p/q) and (x*q = p*y) then Set n: = x*q
Step 5: Set key_arr []: = n
Step 6: if (length of text = odd number) Then text: = text + @
Step 7: Set i: = 0, c: = “”, j: = 0
Step 8: while (i < length of text) Repeat
Step 9: if in = EVEN number Set c: = c+text [i] + (m*key_arr[j]) Else
Step 10: Set c: = c+text [i] – (m*key_arr[j]) End if
Step 11: Set j: = j+1
Step 12: if j = key_length then set j: = 0
Step 13: Print c
Step 14: Stop

- **Heuristic encryption:**
  A genetic algorithm is being applied:
  Input: Intermediate Cipher (c)
  Output: Final Cipher

  Step 1: Set st1: = substring of c (1 to c/2)
  Step 2: substring of c (c/2 to c)
  Step 3: while (i<length of text) Repeat Step 4 to Step 8
  Step 4: perform crossover at mate point I with st1 (0, i) and st 2 (I, n) and st 2 (0, i) and st 1 (I, n)
  Step 5: Set st: = st 1 + Reserve (st 2)
  Step 6: Reserve st and set gst = st
  Step 7: compare c and gst to find out fit value
  Step 8: select the gst having minimum no of fit value
  Step 9: Return gst
  Step 10: Stop

- **Decoding heuristic algorithm:**
  Implementation of the genetic algorithm:
  Input: Final Cipher (c)
  Output: Intermediate Cipher
  Step 1: Set st: = reserve (cipher)
  Step 2: Set n = length of cipher
  Step 3: Set st 1: = substring of c (1 to n/2) st 2: = substring of c (n/2 to n)
  Step 4: Set st 2: = Reverse of st 2 i: = 1
  Step 5: while (i<length of st 2) Repeat Step 6 to Step 9
  Step 6: perform ctossover at mate point i with st 1 (0, i) and st 2 (i, n) and st 2 (0, i) and st 1 (i, n)
  Step 7: Set gst: = st 1 + st 2
  Step 8: compare gst and cipher to find out fit value (no of character position Remain unchanged)
  Step 9: select the gst having minimum no fit value
  Step 10: Return gst
  Step 11: Stop

- **Decryption heuristic algorithm with an example:**
  Function:
  Input: Intermediate cipher, keytype, public key (p, q) or private key (x, y)
  Output: Plain Text
  Step 1: if key type = public key Set m: = 2 and Read private key (x, y) from atabase else Set m: = 1 and Read public key (p, q) from database
  Step 2: Set x: = ((x-(y-1))*y)+y
  Step 3: Set p: = ((p-(q-1))*q)+q
  Step 4: if (x/y = p/q) and (x*q = p*y) then Set n:=x*q
  Step 5: Set key_arr [] : = n
  Step 6: Set i = 0, c = "", j = 0
  Step 7: while (i<length of plmcipher) Repeat Step 4 to Step 7
  Step 8: if i = EVEN number then Set c: = c + plmcipher [i] – (m*key [j]) Else
  Step 9: Set c: = c + plmsiphert [i] + (m*key [j]) End if
  Step 10: Set j: = j + 1

  Step 11: if j = length of key [] then Set j: = 0
  Step 12: Print c

- **Cryptographic algorithm:**
  Step 1: Start
  Step 2: Call Proposed Encryption Heuristic
  Step 3: Call Proposed Encryption Heuristic using Genetic Algorithm
  Step 4: Call Proposed Encryption Heuristic using Genetic Algorithm
  Step 5: Call Proposed Decryption Heuristic
  Step 6: Stop

3.6. Conclusions

The developed encryption algorithm, enhanced with the operators of the genetic algorithm, generates code ciphers with a high degree of entropy, estimated by the minimum number of matching positions of

the characters in the encrypted message, which provides a high degree of protection for text transmitted over the computer network.
Through the developed encryption algorithm with randomly selected two mutually non-prime numbers, enhanced with the operators of the genetic algorithm, it can be interpreted as an extension of the RSA encryption algorithm's scope of application, implemented by C # software tools.
The encryption algorithm developed through the application of the operators of the genetic algorithm reaffirms the encryption properties of known similar algorithms by developing a new software product.
The genetic algorithm developed and implemented in the C # environment has reliable and stable encryption with a high degree of protection for the transmitted message and can be applied in data exchange requiring a high level of security.

CHAPTER IV
CONCLUSION-SUMMARY OF THE RESULTS OBTAINED

In conclusion, it should be noted that in accordance with the purpose and tasks assigned by the dissertation paper mathematical modeling of processes in computer networks under the influence of malware, defining the syntax of a genetic algorithm for detecting computer network intrusions have been performed, a modified encryption algorithm enhanced by a genetic algorithm when constructing the encrypted message has been proposed.
The following scientific, applied-scientific, and applied results have been obtained, which are presented in a summary according to the requirements of Art. 27 (2) of the Regulations for application of ADASRB:

4.1. SCIENTIFIC RESULTS

4.1.1. A model of the processes of susceptibility, exposure, infection and recovery of a computer network in case of malware exposure is proposed, described by a system of differential equations for instantaneous and prognostic estimation of the computer network - a system of equations (6).

4.1.2. An original solution of the system of differential equations is proposed in two cases - equilibrium for constant variables - analytical expression (37) - Chapter I and lack of equilibrium for time-dependent variables - analytical expression (65) - Chapter I, defining machine state classes in the computer network.
4.1.3. Analytical expressions for calculating the network characteristics in case of susceptibility, exposure, infection, and recovery (reconstruction) of machines in the computer network during an attack with malicious software are derived фrom the solution of the system of inhomogeneous differential equations.

4.2. SCIENTIFIC APPLIED RESULTS

4.2.1. A software tool has been developed to protect the computer network by encrypting the transmitted information using the operators of a genetic algorithm implemented with the C # programming language (Chapter III).
4.2.2. Software products implemented in the Matlab environment have been developed to illustrate the solutions of the system of differential equations for instantaneous and prognostic estimation.

4.3. APPLIED RESULTS

4.3.1. The list of attributes, gene structure, and codes of the network characteristics of the chromosome (rule) in the genetic algorithm has been expanded, expanding the scope of its action to detect network intrusions (Table 1, Chapter II).
4.3.2. The structure of the fitness function of Firas Alabsi is evaluated and the basic operations on chromosomes of a given generation are illustrated with data obtained from a numerical experiment. 4.3.3. The results of the evaluation are shown in Fig. 5 (Chapter III).
Experimental evaluation of the parameters A, and AB of the fitness function of Firas Alabsi is made with data obtained from simulation of network communications of the type Normal, DoS, R2L, U2R, and Probe, realized by randomly generated network characteristics of five chromosomal structures for each category. The results are presented in Table 2(Chapter III).

4.3.4. These scientific, scientific-applied and applied results are a good basis and prerequisite for future research in technology: Industry 4.0-artificial intelligence, Internet of Things, robotics, cybersecurity, as well as decision-making methods, multicriteria synthesis, analysis, prevention and risk management, big data, fuzzy evaluations and Soft Computing.

Certain concentrated efforts and creative approaches are needed in training with an emphasis on e-learning to create knowledge, competencies, and skills in a wide range of students, researchers and professionals.

LIST OF AUTHOR'S PUBLICATIONS ON THE DISSERTATION TOPIC

1. Lazarov, A., P. Petrova. Genetic algorithm in computer network protection. Engineering Sciences, No. 1, pp. 80-95, ISSN: 1312-5702. E-ISSN: 2003-3542, DOI: 10.7546/EngSci.LIX.22.01.07, LIX, 2022.

2. Lazarov, A., P. Petrova, Modelling activity of a malicious user in Computer Networks. Cybernetics and information technologies, Volume, No 2, Sofia 2022. Print ISSN: 1311-9702. On-line ISSN: 1314-4081. SJR 027, Q2.

3. Lazarov, A., P. Petrova. Crypto genetic approach in information security, XXII International Symposium on Electrical Apparatus and Technologies SIELA 2022, 1-4 June 2022, Burgas, Bulgaria (in print).

4. Лазаров, А., П. Петрова. Концепция на генетичния алгоритъм за откриване на прониквания в компютърната мрежа, Електронно списание на център по Информатика и технически науки на Бургаски Свободен Университет, Том 8, бр. 1, 2019, pp. 3-12.

5. Лазаров, А., П. Петрова. Определяне на функцията на пригодност в генетичния алгоритъм за откриване на проникване в компютърните мрежи, Списание „Компютърни науки и комуникации", Том 8, No1 (2019), БСУ, Бургас, pp. 13-22.

6. Лазаров, А., П. Петрова. Моделиране на процесите при въздействие на компютърна мрежа със злонамерен софтуер, Бургаски свободен университет, ЦИТН, Годишник на БСУ том XXXVIII, 2018, pp. 5-22.

THANKSGIVING

The management and lecturers from the Center for Informatics and Technical Sciences of BFU and my supervisors: Prof. DSCTECH Andon Dimitrov Lazarov from VVMU "N. Y. Vaptsarov" and Prof. Dr. Georgi Georgiev Dimitrov from University of Library Science and Information Technology played definitely a significant role in the development and results of this dissertation.

The critical concrete remarks and constructive proposals of Assoc. Prof. Dr. Penka Georgieva and Assoc. Prof. Dr. Veselina Zhecheva have their contribution to the completed form and content of the dissertation.

My sincere thanks to everyone!

Thank you to my family for their full selfless support and empathy.

PETYA PETROVA