Neural Network Cryptographic Obfuscation for Trusted Cloud Computing

Vladimir L. Eliseev[1,2], Ekaterina A. Miliukova[2] and Sergey V. Kolpinskiy[1,2]

[1] JSC InfoTeCS, Otradnaya st. 2B building 1, Moscow, 127273, Russia
[2] Moscow Power Engineering Institute, Krasnokazarmennaya st. 14, Moscow, 111250, Russia

Abstract

A problem of trusted computation in the paradigm of cloud processing system and the related adversary model are discussed. Obfuscation, traditional and homomorphic cryptography are reviewed as candidate technologies. An approach to create a trusted cloud computing system with the help of neural network obfuscation of encrypted data processing is introduced.

Keywords

Artificial Neural Networks, Obfuscation, Encrypted Data Processing, Cloud Computing, Homomorphic Encryption

1. Introduction

The development of modern computer technology and telecommunications has led to a significant change in the paradigm of data processing. Cloud computing has become widespread, in which data processing is carried out remotely in large computing centers, and the sources of data are computers and mobile devices of users. Data processing centers (DPCs), which are the backbone of cloud computing infrastructure, function by sharing and virtualizing resources - network addresses, traffic, processors, memory, software, etc. Trust in the cloud provider is a cornerstone as the algorithms and data being processed reside in an infrastructure that they control completely.

Data processing algorithms in often represent objects of intellectual property and sometimes trade secrets, so they should not be available for study by outsiders. Examples are banking and insurance scoring algorithms, checking software licensing conditions, and some applied know-how, for example, in the analysis of medical and marketing data.

The data itself, processed in cloud data centers, in the overwhelming majority of cases is also a value that requires one degree or another of protection. It can be personal data of people, including biometric and medical information, as well as commercially sensitive information. Currently, cryptographic data protection, which guarantees confidentiality, integrity and authenticity, is provided only during data transmission, but not during their processing.

In connection with all of the above, it seems relevant to develop an approach that would ensure the protection of both algorithms and the data processed by it, as well as be quite effective in real operation in an untrusted environment of a cloud agent.

2. Threat and intruder model

When considering the issue of trusted data processing in a public cloud, it is necessary to determine the location of a potential intruder and his opportunities to attack the transmitted and processed data. The basic attacker model, widely considered, is to be located outside the physical infrastructure of a cloud data center. A cloud provider as a legal entity, its infrastructure and service personnel in this case

Russian Advances in Fuzzy Systems and Soft Computing: Selected Contributions to the 10th International Conference «Integrated Models and Soft Computing in Artificial Intelligence» (IMSC-2021), May 17–20, 2021, Kolomna, Russian Federation
EMAIL: vlad-eliseev@mail.ru (A. 1); k.milyush@bk.ru (A. 2); skolpinsky@bk.ru (A. 3)
ORCID: 0000-0002-9341-7475 (A. 1)
© 2021 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

CEUR-WS.org/Vol-2965/paper26.pdf

are considered as a completely trusted environment for the user's programs and data. An external attacker can attack data sent to and from the cloud (Figure 1). To ensure confidentiality, integrity and authenticity of data, in this case, it is appropriate to use cryptographic protocols (IPsec, TLS, etc.). However, in fact, the public cloud is the place of presence of many users, including an intruder.

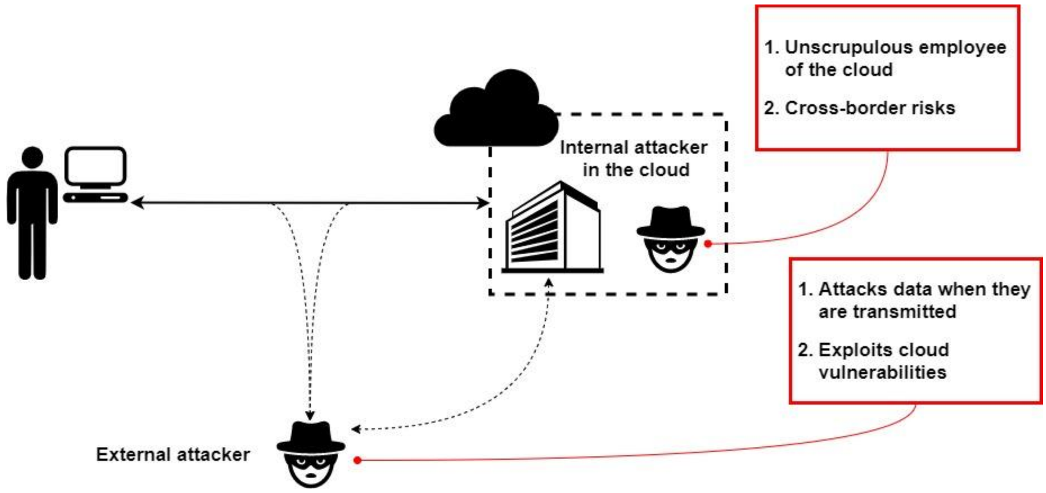
1. Unscrupulous employee of the cloud
2. Cross-border risks
Internal attacker in the cloud
1. Attacks data when they are transmitted
2. Exploits cloud vulnerabilities
External attacker

Figure 1: A model of an intruder when processing data in the cloud

To increase confidence in cloud computing, the intruder's baseline model includes the possibility of its presence in the data center infrastructure. Traditional security measures in this case include identification and access management (IAM), which allows you to isolate users within their own domains of cloud resources. This does not guarantee absolute safety. In particular, OAuth authentication tokens, which are widely used in cloud systems, if stolen, open the attacker's access to the user's cloud resources. This attack is known as Man-in-the-Cloud, but the attacker model under consideration could be considered quite satisfactory until the discovery of vulnerabilities in modern processors, attacks on which were named Specter and Meltdown. These attacks, under some conditions, allow you to read the memory of other processes and even the operating system kernel, bypassing the IAM restrictions. The only effective way to avoid Specter and Meltdown attacks is to run user programs on a dedicated server only for them, which effectively destroys the very foundation of the cost-effectiveness of the cloud computing model.

The intruder model, which allows him to have administrator rights for parts of the data center infrastructure, requires a deeply thought out IAM system for data center personnel and significant resources to support it. At the same time, there remains the risk of negligence or collusion of data center employees for the purpose of coordinated actions to attack the user's cloud resources.

One of the challenges of cloud computing in today's world is cross-border risks due to the fact that the user and the physical infrastructure of the cloud are often located in different jurisdictions. The perfection of cloud technologies makes it possible not to notice where data is being processed now: in the USA, Europe or on the territory of Russia, however, from the point of view of the legislation of most developed countries of the world, there are categories of data, the movement of which abroad, even in encrypted form, is prohibited. Obviously, this prohibition is due to the fact that the movement of information across the border for processing in the cloud makes it available to foreign countries, contrary to national interests.

3. Overview and analysis of the protection technologies

3.1. Obfuscation

To preserve the secrecy of algorithms, methods of obfuscation of the program code can be used. Obfuscation (hiding) is the process of transforming an algorithm into a form that preserves its functionality, but makes it difficult to understand its work. For obfuscation, numerous heuristic methods have been developed that automate the equivalent transformation of the program in order to obfuscate the original structure of the algorithm by changing the representation of the data and constants used. Such a transformation gives visually impressive results, however, for each of the heuristic obfuscating algorithms, an inverse algorithm can be created that restores the original structure. Therefore, heuristic obfuscation cannot be considered a reliable approach to protect algorithms containing confidential information.

The problem of hiding the logic of a program was first formulated in the 1970s, but before the appearance of [1], all obfuscation methods were heuristics that were difficult to evaluate and compare with each other. In [1], the strict notion of persistent obfuscation was introduced for the first time, but it was demonstrated that complete concealment of the program logic is impossible. In [2], it was proved that the best possible obfuscator is a fuzzy obfuscator, that is, one that reports no more about the original program than any other program with functionality equivalent to the original.

In 2013, the first obfuscation method was presented that provably implements the fuzzy property called graded encoding [3]. The implementation of this approach for applied applications is significantly inefficient from a computational point of view. In 2019, artificial neural networks were proposed as an obfuscator of indistinguishability [4], the application of which is discussed in detail in Section 4.

3.2. Traditional cryptography

The generally accepted approach for data protection in modern computer systems and data transmission networks is the use of cryptographic algorithms and protocols. Among them, there are groups of symmetric and asymmetric cryptographic algorithms.

Symmetric algorithms are based on a secret key, which must be possessed by all parties to the information exchange. With regard to cloud systems, this means that in the cloud infrastructure, along with algorithms for applied data processing, an encryption key must also be present for receiving and transmitting data over the network. Obviously, in the case of an untrusted cloud infrastructure, the encryption key does not give reason to consider the data you are processing in clear text as protected.

Asymmetric algorithms are based on dividing the unique key of each side of the information exchange into two parts: public and private. The private key must be kept secret, and it provides proof of ownership of the public key, which is freely available. Using the asymmetric Diffie-Hellman protocol, two parties to an information exchange can generate a shared secret key that is unknown to an attacker who does not have access to the subscribers' private keys. However, this secret key will not be used in any case to protect the processed data, which is equivalent to the situation with a symmetric secret key.

3.3. Homomorphic cryptography

Currently, the only approach that ensures the execution of trusted computations in an untrusted environment is homomorphic cryptography, which implies performing operations on encrypted data, while the operations themselves are implemented in a way that does not depend on the encryption key. The homomorphic encryption algorithm developed in 2009 was extremely ineffective: compared to conventional computations, the performance decreased by a factor of 1012 [5]. Continuing research has allowed one to two orders of magnitude to improve performance [6], but still far from practically useful applications.

In [7], an approach was proposed and tested for processing encrypted medical data using neural networks. Using this approach, 99% accuracy of classification of handwritten characters over encrypted MNIST data has been demonstrated [8]. It was noted that an additional problem is the slowdown in learning when using encrypted data.

The prospects for homomorphic cryptography continue to be assessed quite high, which is confirmed by both research funding in the largest corporations (IBM, Microsoft, Intel, Google) and the emergence of startups (Duality, Enveil, CryptoNext Security) [9].

Thus, we can state that modern cloud technologies carry many risks for user data, who would like to use public clouds to process it. An effective homomorphic cryptographic system is the cardinal way of resolving risks; however, as shown above, the degree of maturity of this approach is not yet sufficient for practically useful applications.

4. Neural network cryptographic obfuscation

The technologies discussed above cannot currently fully solve the problem of secure data processing in an untrusted cloud environment. At the same time, a potential solution to this problem can be the use of artificial neural networks both as a mechanism for obfuscating algorithms and as a mechanism for processing encrypted data. Together, this can provide properties that resemble homomorphic encryption, but based on artificial neural networks as an obfuscation tool.

4.1. Artificial neural networks

Hecht-Nielsen in 1987 based on the works of A.N. Kolmogorov proved the representability of any continuous function of many variables using a two-layer multilayer perceptron (MLP) with any predetermined accuracy. To synthesize an artificial neural network (ANN) for a specific applied problem, an optimization process called learning is usually used.

The difficulties in extracting knowledge from a trained neural network are well known, in connection with which the question is posed in [10], can they be considered a black box? The lack of a clear understanding of the internal mechanisms of ANN even called into question the possibility of their use in critical applied areas [11]. This feature can be used for the good, hiding in the ANN information about the actions it performs, as well as for working with encrypted data.

The idea of encrypted data processing using neural networks was formulated in different ways in [10] and [11], however, intermediate formal constructions in the form of polynomials and homomorphic cryptographic algorithms were proposed. This indicates that the authors of these works underestimate the capabilities of neural networks for processing encrypted data.

4.2. Neural network obfuscation

Consider the definitions of strict obfuscation and indistinguishability obfuscation according to [1].

By strict obfuscation we mean such transformation of the program that the functioning of the obfuscated program is completely equivalent to the original one, but the only way to understand what the obfuscated program does is to run it multiple times. It has been shown that there is a class of programs for which strict obfuscation is unattainable.

Indistinguishability obfuscation is such a transformation of a program that, being carried out for two functionally equivalent different programs, it is impossible to understand which of the obfuscated programs corresponds to one of the two original ones.

Consider the problem of obfuscating the program P using ANN. We define P in general form as a deterministic Boolean function of a vector of variables x from the set of admissible vectors $X \subseteq \mathbb{B}^n$, which calculates the vector of values y from the set $Y \subseteq \mathbb{B}^m$:

$$y = P(x) \ \mid x \in X, y \in Y$$

A neural network obfuscator of a vector Boolean function P is a neural network N with the number of inputs n and the number of outputs m, such that

$$\forall x \in X : y = P(x), \hat{y} = \mathcal{N}(x), |y_i - \hat{y}_i| < 0.5 \quad i = \overline{1, m}$$

The synthesis of a neural network $\mathcal{N}$, satisfying the definition of a neural network obfuscator, can be carried out on a training set of M pairs (x_j, y_j), where $j = \overline{1, M}$. Since the dimension of the neural network output is m Boolean values, then $M \leq 2^m$.

In [4], theorems on functionality and indistinguishability of a neural network obfuscator of a Boolean function were proved.

Thus, neural networks can be used to hide algorithms within the structure of weights. This can be shown by the example of an algorithm that implements operations on integers and in which individual arithmetic calculations are replaced by an equivalent neural network implementation. An example of such an approach is demonstrated for a simplified bank scoring algorithm in [14]. In order to make the technique more practical and convenient for applied applications, in [15], refinements were introduced regarding the introduction of general requirements for functions for which neural network analogs are constructed.

It is well known that the synthesis of the architecture of artificial neural networks for specific applied problems is a poorly researched area in which heuristic approaches are often used and the developer's experience is of great importance. It is necessary to make sure that it is practically possible to synthesize neural networks for calculating Boolean functions. In this case, it is necessary to investigate to what extent the structure of the ANN and the complexity of its synthesis are related to the complexity of the Boolean function. As examples of Boolean functions, one can consider arithmetic operations, random maps, affine and bent functions. Corresponding studies were carried out in [16] and [17], and in the last work, special attention was paid to the study of the relationship between the complexity of a Boolean function and the complexity of an equivalent neural network. Under the complexity of the neural network, it is logical to take the number of weight coefficients. Various approaches can be used as a measure of the complexity of a Boolean function. In [17], the degree of nonlinearity of the Boolean function was chosen. The results of several experiments have shown that for linear and bent functions, the complexity of the equivalent ANN is approximately the same.

4.3. Encrypted data processing

Based on the results obtained on neural network obfuscation, as well as the capabilities of the ANN to approximate arbitrary functions, it is possible to set the task of processing encrypted data. As a cipher, we will use tables of random substitutions as the most general case for encrypting the input and output data of the ANN. In order for the algorithm to be presented according to the method described in [14], it is necessary to adopt it for the synthesis of ANNs working on encrypted data. We consider to use pairs $\left(E_x(x_j), E_y(y_j)\right)$ instead of training pairs (x_j, y_j) for training a neural network, where E_x, E_y are encryption operations using a table of random substitutions of input and output Boolean vectors of the original function and the neural network replacing it. In [18], it was shown that for the size of the input vector $n \geq 6$, the brute-force attack complexity for a random substitution cipher, which is a common case of an encrypted input of a neural network cryptographic obfuscator, exceeds the the brute-force attack for modern ciphers with a 256-bit key since $(2^n)! > 2^k$ satisfies for $k = 256$.

The theoretical results obtained in [18] made it possible to demonstrate the practical application of the ANN for processing encrypted data [19]. For three functions of different classes, experiments were carried out with the creation of equivalent neural networks that process open and encrypted data. For encryption, tables of random substitutions were used. Studies have shown that neural networks processing encrypted data require more neurons in the hidden layer, that is, in the general case, the metric of computational complexity for encrypted data is higher than for open data. In general, training neural networks to process encrypted data is more time consuming than for open data.

The overhead costs of processing encrypted data by neural networks can be divided into two stages: training the ANN and directly processing the encrypted data. As already noted, training is a rather lengthy and resource-intensive process and should be carried out in a trusted environment, since it is built using lookup tables, which are the encryption key. The computations themselves, carried out by the already trained neural network over the encrypted data, are no longer exponentially complex and are proportional to the number of ANN weight coefficients.

On the basis of the considered approaches to the use of neural networks for cryptographic obfuscation, it is proposed to develop a trusted cloud computing system [20]. The system should consist of three subsystems (Figure 2): preparation of neural network algorithms, data exchange with the cloud, and cloud computing agents. It is assumed that when implementing a full cycle of actions in the system to prepare a certain computational algorithm for operation in an untrusted computing environment, a potential attacker in a cloud computing environment will have only a cryptographically obfuscated algorithm. The encryption keys and algorithm will not be transferred to the cloud in the clear. The input data for the cryptographically obfuscated algorithm placed in the cloud will be encrypted before being sent over the Internet, and the encrypted results will also be decrypted in the user's local trusted environment. Thus, the designed system allows trusted calculations in the infrastructure of the cloud agent without the need to trust him.

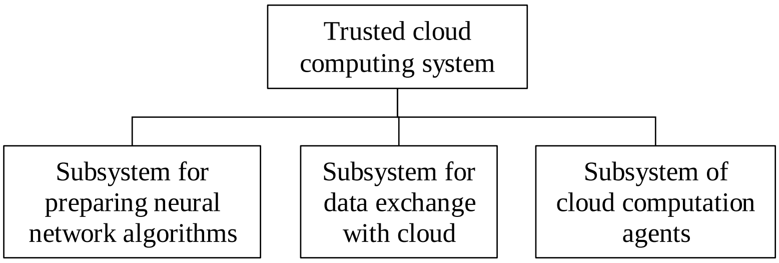
Trusted cloud computing system
Subsystem for preparing neural network algorithms
Subsystem for data exchange with cloud
Subsystem of cloud computation agents

Figure 2: Composition of a trusted cloud computing system

The proposed approach indivisibly combines encryption and obfuscation, protecting the confidentiality of both the data and the algorithm that processes it. At the same time, the proposed approach satisfies the well-known Kerkhoffs principle, since the knowledge of the encryption algorithm by an attacker, for example, the fact of using a table of random substitutions, does not make the proposed approach insecure. Similarly, the knowledge of the neural network training principles by the attacker and full access to the neural network implementation of the algorithm does not make encryption and obfuscation less secure.

5. Discussion

Analyzing the proposed approach for neural network processing of encrypted data, the following advantages should be noted:

- high performance when performing obfuscated algorithms versus homomorphic cryptography methods;
- in the case of using tables of random permutations, the security has a factorial estimate of the enumeration complexity;
- in addition to data protection, the protection of the calculation algorithm is provided.

At the same time, this approach requires further research, since some of its shortcomings prevent widespread adoption:

- high computational complexity of learning neural networks;
- there is no methodology for substantiated synthesis of neural network architecture;
- application for data encryption using random substitution tables is susceptible to attacks using frequency analysis;
- an issue of the loss of security and obfuscation in the case of overfitting remains open.

6. Conclusion

The paper presents an analysis of the problems of providing trusted cloud computing. Various approaches to solving this problem are considered, and an approach based on artificial neural networks is proposed and substantiated.

7. References

[1] Barak B., Goldreich O., Impagliazzo R., Rudich S., Sahai A., Vadhan S., Ke Yang. On the (im)possibility of obfuscating programs. J. Kilian (ed.) // Advances in Cryptology — Crypto’01. LNCS, v.2139, Springer-Verlag, 2001, pp. 1–18.

[2] Garg S., Gentry C., Halevi S., Raykova M., Sahai A., Waters B. Candidate indistinguishability obfuscation and functional encryption for all circuits // Proc. of the 54th IEEE Annual Symposium on Foundations of Computer Science, 2013, pp. 40–49. doi:10.1109/FOCS.2013.13.

[3] M. R. Albrecht, C. Cocis, F. Laguillaumie, A. Langlois Implementing Candidate Graded Encoding Schemes from Ideal Lattices // International Conference on the Theory and Application of Cryptology and Information Security, ASIACRYPT 2015, pp. 752–775.

[4] Eliseev V.L. Iskusstvennye nejronnye seti kak mehanizm obfuskacii vychislenij // Trudy vserossijskoj konferencii «XVIII Sibirskaja nauchnaja shkola-seminar s mezhdunarodnym uchastiem «Komp'juternaja bezopasnost' i kriptografija» – SIBECRYPT'19», Tomsk, 9–14 sentjabrja 2019 g., Prikladnaja diskretnaja matematika. Prilozhenie, 2019, №12, s.165–169, doi: 10.17223/2226308X/12/46. [In Russian]

[5] Microsoft researchers smash homomorphic encryption speed barrier. URL: https://www.theregister.com/2016/02/09/researchers_break_homomorphic_encryption/.

[6] S. Halevi, V. Shoup. Faster Homomorphic Linear Transformations in HElib // Cryptology ePrint Archive, Report 2018/244, 2018.

[7] Graepel T., Lauter K., Naehrig M. ML Confidential: Machine Learning on Encrypted Data // Information Security and Cryptology – ICISC 2012. Lecture Notes in Computer Science, vol 7839. Springer, Berlin, Heidelberg, 2012, pp.1-21. doi: 10.1007/978-3-642-37682-5_1.

[8] Gilad-Bachrach, R., Dowlin, N., Laine, K., Lauter, K., Naehrig, M. & Wernsing, J. CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy // Proceedings of The 33rd International Conference on Machine Learning, in Proceedings of Machine Learning Research 48:201-210, 2016.

[9] Duality, a security startup co-founded by the creator of homomorphic encryption, raises $16M. URL: https://techcrunch.com/2019/10/30/duality-cybersecurity-16-million/.

[10] J. M. Benitez, J. L. Castro, I. Requena Are artificial neural networks black boxes? // Trans. Neur. Netw. 8, 5 (September 1997), pp.1156-1164.

[11] J.T. Yao Knowledge Extracted from Trained Neural Networks What's next? // Department of computer Sciences University of Ragina Canada, 2006.

[12] Xie, P., Bilenko, M., Finley, T., Gilad-Bachrach, R., Lauter, K., Naehrig, M. Crypto-nets: Neural networks over encrypted data // arXiv preprint arXiv:1412.6181, 2014.

[13] R. Dathathri, O. Saarikivi, H. Chen, K. Laine, K. Lauter, S. Maleki, M. Musuvathi, T. Mytkowicz CHET: an optimizing compiler for fully-homomorphic neural-network inferencing // In Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2019). ACM, New York, NY, USA, 142-156. doi: 10.1145/3314221.3314628.

[14] Kolpinskij S.V., Miljukova E.A., Eliseev V.L. Nejrosetevaja obfuskacija algoritma bankovskogo skoringa // Sbornik materialov XXV Mezhdunarodnoj nauchno-tehnicheskoj konferencii «Informacionnye sistemy i tehnologii – 2019», Nizhnij Novgorod, 19 aprelja 2019 g., s.725–732. [In Russian]

[15] Miljukova E.A., Eliseev V.L. Razrabotka sposoba predstavlenija vychislitel'nogo algoritma dlja nejrosetevoj obfuskacii // Tezisy dokladov XXVII Mezhdunarodnoj nauchno-tehnicheskoj konferencii studentov i aspirantov «Radiojelektronika, jelektrotehnika i jenergetika», Moskva, 11–12 marta 2021, s.274. [In Russian]

[16] Miljukova E.A., Eliseev V.L. Issledovanie slozhnosti nejrosetevoj obfuskacii vektornyh bulevyh funkcij // Sbornik trudov XXVIII Mezhdunarodnoj nauchno-tehnicheskoj konferencii "Sovremennye tehnologii v zadachah upravlenija, avtomatiki i obrabotki informacii", Alushta, 14–20 sentjabrja 2019 g., s.75–76. [In Russian]

[17] Miljukova E.A., Eliseev V.L. Nejrosetevaja realizacija nekotoryh klassov bulevyh funkcij // Sbornik materialov XXVI Mezhdunarodnoj nauchno-tehnicheskoj konferencii «Informacionnye sistemy i tehnologii – 2020», Nizhnij Novgorod, 24,27,28 aprelja 2020 g., s.786–797. [In Russian]

[18] Eliseev V.L. Nejrosetevaja obfuskacija vychislenij nad zashifrovannymi dannymi // Prikladnaja diskretnaja matematika. Prilozhenie, 2020, №13, s.85–93. doi: 10.17223/2226308X/13/25. [In Russian]

[19] Miljukova E.A., Eliseev V.L. Issledovanie nejrosetevoj obfuskacii dlja obrabotki konfidencial'nyh dannyh // [in press] [In Russian]

[20] Miljukova E.A., Eliseev V.L. Proektirovanie sistemy doverennyh oblachnyh vychislenij s ispol'zovaniem nejrosetevoj obfuskacii // Sbornik trudov XXIX Mezhdunarodnoj nauchno-tehnicheskoj konferencii «Sovremennye tehnologii v zadachah upravlenija, avtomatiki i obrabotki informacii», Alushta, 14–20 sentjabrja 2020 g., s.166–167. [In Russian]