Pseudorandom Sequences Classification Algorithm

Alexander Kozachok, Andrey Spirin

***Abstract*—Currently, the number of information leaks caused by internal violators has increased. One of the possible channels for information leaks is the transmission of data in encrypted or compressed form, since modern DLP (data leakage prevention) systems are not able to detect signatures and other information related to confidential information in such data. The article presents an algorithm for classifying sequences formed by encryption and compression algorithms. An array of frequencies of occurrence of binary subsequences of length *N* bits was used as a feature space. File headers or any other contextual information were not used to construct the feature space. The presented algorithm has shown the accuracy of classification of the sequences specified in the work 0.98 and can be implemented in DLP systems to prevent the transmission of information in encrypted or compressed form.**

***Tóm tắt*—Hiện nay, số vụ rò rỉ thông tin bởi đối tượng vi phạm trong nội bộ gây ra ngày càng gia tăng. Một trong những kênh có thể dẫn đến rò rỉ thông tin là việc truyền dữ liệu ở dạng mã hóa hoặc nén, vì các hệ thống chống rò rỉ dữ liệu (DLP) hiện đại không thể phát hiện chữ ký và thông tin trong loại dữ liệu này. Nội dung bài báo trình bày thuật toán phân loại các chuỗi được hình thành bằng thuật toán mã hóa và nén. Một mảng tần số xuất hiện của các chuỗi con nhị phân có độ dài *N* bit được sử dụng làm không gian đặc trưng. Tiêu đề tệp hoặc bất kỳ thông tin ngữ cảnh nào khác không được sử dụng để xây dựng không gian đối tượng. Thuật toán được trình bày có độ chính xác trong việc phân loại các chuỗi đạt 0,98 và có thể được áp dụng trong các hệ thống DLP để ngăn chặn việc rò rỉ thông tin khi truyền thông tin ở dạng mã hóa hoặc nén.**

Keywords—statistical data analysis, machine learning, classification of binary sequences, DLP systems, information leakage protection.

Từ khóa—phân tích dữ liệu thống kê, học máy, phân loại chuỗi nhị phân, hệ thống DLP, bảo vệ chống rò rỉ thông tin.

This manuscript is received on October 12, 2020. It is commented on November 18, 2020 and is accepted on January 4, 2021 by the first reviewer. It is commented on December 6, 2020 and is accepted on January 10, 2021 by the second reviewer.

I. Introduction

Recently, according to the reports of information and analytical agencies, the number of incidents related to information leaks caused by internal violators has increased [1].

In [2], [3], it is noted that the reasons for the leakage of confidential data can be various factors: the widespread use of information technologies in almost all processes of data processing and transmission, the introduction of remote workplaces, insufficient training of employees in the field of information security, non-compliance with a set of organizational measures, etc. Internal violations pose the greatest threat, since their actions are not analyzed by means of protection aimed at preventing external attacks. Internal violators are cut off mainly by DLP (data leakage prevention) systems.

In [4], [5], it is noted that data protection from internal intruders is a complex task, which is confirmed by the absence of mechanisms for analyzing encrypted or compressed data in DLP systems, in the absence of information about the compression algorithm [6], [7].

In [8], the authors distinguish 2 groups of methods used in DLP systems: content-based and contextual. Content methods use semantic analysis of transmitted information, signature search, and search for digital casts and regular expressions to detect confidential data [9]-[12]. Contextual methods use metadata [13]. In [14], the authors suggest using behavioral methods that generate patterns of standard actions of users or processes when working with data that will differ from the actions of violators.

In [15], the authors consider a method for preventing information leaks based on contextual integrity. The method is based on the idea of legitimate information flows.

In [16]-[20], methods for identifying crypto algorithms in various modes of operation are considered. Classifiers trained on feature spaces

formed during the execution of sub-accounts of the frequency of occurrence of various character sequences, bytes or bits are one of the solutions to the problem of identifying crypto algorithms.

In [21], convolutional neural networks GoogleNet and AlexNet are used for binary classification of AES and DES encryption algorithms in the mode of simple substitution and simple substitution with gearing. Both networks showed high classification results with an accuracy of more than 0.9, but the GoogleNet network has higher accuracy values on some pairs of cryptographic algorithms.

A similar problem of classification of harmful traffic by machine learning methods was solved in [22]-[26]. In [22]-[24], were used methods based on convolutional neural networks, the main advantage of which, in comparison with standard machine learning algorithms, is that there is no need to search for and construct a feature space explicitly. In [25], the authors proposed using a combination of machine learning algorithms with and without a teacher to overcome the zero-day vulnerability when a previously unknown type of attack occurs. The paper [26] provides an overview of machine learning methods used for classifying traffic, describes the stages of training and building classifiers.

In [27], a comparative analysis of machine learning methods based on neural networks for classification of encrypted and compressed data is performed. The convolutional neural network showed the highest accuracy of 0.669, the sequential neural network showed an accuracy of 0.541, and the k-nearest neighbor method showed 0.6. These results allow us to conclude that it is necessary to study the applicability of other machine learning methods to solve the problem of classification of encrypted and compressed data.

In work [22], it is noted that the growth of Internet traffic and the increasing number of devices that generate it create a certain complexity for DLP systems. Modern traffic filtering systems cannot accurately and effectively detect information with high entropy, such as encrypted and compressed data, which makes the developed model relevant.

In work [23], it is noted that existing classifiers can hardly cope with the task of classifying encrypted and compressed data. The authors propose an algorithm for feature extraction based on calculating the entropy of the packet data content. The method is based on increasing message redundancy by generating new binary strings from the analyzed data. To form a feature space, the authors propose to form a matrix of size 8*4, the rows of which are the step value, when forming redundant binary rows, and the columns are the values of binary subsequences, for which the entropy value is calculated in the obtained data. The generated feature space is used for training classifiers based on the method of support vectors or a random forest. The results obtained by the authors indicate a significant influence of the data type on the classification results. The worst classification accuracy values were obtained for audio files (0.65), for video files the classification accuracy value was less than 0.7, and for images and text – approximately 0.72.

Based on the analysis of the literature, we can conclude that the classification of encrypted and compressed data is not sufficiently accurate. In our study, we propose an algorithm for extracting features from the analyzed sequence and a classification algorithm based on the ensemble method of constructing a random forest to determine the most significant classification features and then use the decision tree construction algorithm.

II. Algorithm for constructing a feature space

In general, the problem of classification of pseudorandom sequences (PRS) is presented in equation (1) and is formulated as follows: it is necessary to map the original set of PRS X to the set of classes Y based on a classifier trained on the selected sign space.

$$F : X \in \{x_1, \ldots, x_j\} \rightarrow Y \in \{y_1, \ldots, y_i\} \quad (1)$$

where X – the initial set of binary PRSs that are subject to classification, Y – multiple classes, F – classifier display function.

The set of Y classes includes: encrypted and compressed sequences.

To solve the problem of PRS classification, we propose to use an algorithm for extracting features from the analyzed sequence based on statistical approaches: counting the number of sub-sequences of 9 bits in length and the byte distribution. However, 9-bit subsequences imply the presence of 512 statistical features, and the byte distribution assumes the presence of 260 features: 256-byte frequencies and 4 more-the maximum and minimum values of the byte frequency, the mathematical expectation and the standard deviation of the byte distribution. Calculating all the features will require significant processing time for the analyzed data, which in practice means that such an approach is impossible to process data in real time. To overcome this problem at the training stage, it is proposed to use an algorithm for constructing a random forest that allows you to determine the most significant features based on the Gini index or information growth criterion. Based on certain parameters, a classifier is built based on building a decision tree for use in a DLP system.

The PRS classification algorithm consists of 3 stages: the formation of a feature space (*Section II.A*), the construction of a classifier based on it (*Section II.B*), and the application of the resulting classifier to the data under study (*Section III*).

A. Feature space constructing algorithm

The initial data for the algorithm for constructing a feature space are: a class-marked set of PRS with power, and a set of binary subsequences of bit length with power. The set is formed by constructing all possible binary subsequences of a given bit length.

The algorithm for constructing a feature space is shown in Fig. 1.

Data: P: |P|=Q, S: $|S| = 2^{N-1}$
Result: $F_{Q,E}$
1 $F_{Q,E} \leftarrow <>$
2 **for** $p \in P$ **do**
3 $\quad M_p \leftarrow \mathbf{Len}(p)$
4 $\quad$ **for** $s \in S$ **do**
5 $\quad\quad n_s \leftarrow \mathbf{Count}(p,s)$
6 $\quad\quad f_{p,s} \leftarrow \frac{n_s}{M_p - N_s + 1}$
7 $\quad\quad F_{Q,E} \leftarrow F_{Q,E} \cup < f_{p,s}, y_i >$
8 return $F_{Q,E}$

Fig. 1. Features space building algorithm.

Step 1. To initialize an empty tuple of frequencies of the subsequences $F_{Q,E}$.

Step 2. The PRS for each of the plurality of carry:

Determine the length of a sub-sequence and assign its value to a variable M_p.

For each subsequence *s* of the set S execute:

Assign to variable n_s a function value $Count(p,s)$. Function counts the number of occurrences of the subsequence *s* in the PSR p without overlapping.

Assign to variable $f_{p,s}$ the value of the equation (2):

$$\frac{n_s}{(M_p - N_s + 1)}, \qquad (2)$$

where n_s – number of occurrences of the subsequence *s* in the PSR *p* without overlapping, M_p – length of the PSR *p* in bits, N_s – length of the subsequence *s* in bits.

Write to the tuple $F_{Q,E}$ the value of the frequency $f_{p,s}$ and the PSR class y_i.

Step 3. Return tuple $F_{Q,E}$.

The resulting tuple of frequency values of occurrence of bit-length subsequences is a characteristic space for further training and construction of the classifier.

B. PRS Classification

The initial data for performing the PRS classification are: PRS *p*, classifier *K*, set of the features *V*.

The PRS classification algorithm is shown in Fig. 2.

Data: PRS p, classificator <K>,<V>
Result: Class y for PRS

$F_{Q,V} \leftarrow <>$
$State \leftarrow <>$
$M_p \leftarrow \mathbf{Len}(p)$
for $v \in V$ **do**
  $N_v \leftarrow \mathbf{Len}(v)$
  $n_v \leftarrow \mathbf{Count}(p,v)$
  $f_{p,v} = \frac{n_v}{M_p - N_v + 1}$
  $F_{Q,V} = F_{Q,V} \cup f_{p,v}$
$State \leftarrow \mathbf{Next}(k)$
while $State[7] \neq True$ **do**
  if $f_{p,State[2]} \geq State[3]$ **then**
    $State \leftarrow \mathbf{NextRight}(State)$
  else
    $State \leftarrow \mathbf{NextLeft}(State)$
$y_p \leftarrow State[4]$
return y_p

Fig. 2. PRS classification algorithm.

Step 1. Initialize the tuple $F_{Q,V}$ with empty values.

Initialize the tuple *State* with empty values.

Calculate the length M_p of the sequence *p* in bits.

Step 2. For all features *v* from the tuple *V* execute:

Calculate the length of the subsequence *v* and write the resulting value to a variable N_v.

Calculate the number of occurrences of the subsequence *v* in the PRS *p* and write the resulting value to a variable n_v.

Calculate the frequency of occurrence of a subsequence *v* in PRS *p* by equation (2).

Add a value for the frequency of the subsequence *v* in PRS *p* to the tuple $F_{Q,V}$.

III. Experiments

The following sets are used to evaluate the quality of the classifier:

- TP (true positive) – number of correctly classified PRSs belonging to the class $y_i \in Y$.
- TN (true negative) – number of PRSs correctly assigned to a non-class $y_i \in Y$.
- FP (false positive) – the number of PRSs incorrectly assigned to the class $y_i \in Y$, i.e. the number of false positives (the first type of error).
- FN (false negative) – the number of PRSs incorrectly not assigned to the class $y_i \in Y$, i.e. the number of goal skips (second-type error).

To assess the quality of classification, we used the percentage of correct responses metric, which is generally defined by the equation (3).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

For a sample consisting of *K* PRS classes, the percentage of correct answers of the classifier is determined by the equation (4).

$$Accuracy_{total} = \frac{\sum_{i=1}^{K} Accuracy_{y_i}}{K}, \quad (4)$$

where $Accuracy_{y_i}$ – percentage of correct responses for the class y_i.

To determine the percentage of correct responses for each class, the confusion matrix shown in Table I is constructed.

TABLE I. Confusion matrix for classification 4 classes of PSR

		Correct class			
Predicted class	***K***	***1***	***2***	***3***	***4***
	1	T_1	F_{12}	F_{13}	F_{14}
	2	F_{21}	T_2	F_{23}	F_{24}
	3	F_{31}	F_{32}	T_3	F_{33}
	4	F_{41}	F_{42}	F_{43}	T_4

When performing a multi-class classification, sets are calculated based on the error matrix using the following (5):

$$\begin{cases} TP_{y_i} = T_{y_i} \\ TN_{y_i} = \sum_{c=1}^{K} T_c - TP_{y_i} \\ FP_{y_i} = \sum_{c=1}^{K} F_{y_i,c} \\ FN_{y_i} = \sum_{c=1}^{K} F_{c,y_i} \end{cases} \quad (5)$$

where y_i – correct class of PRS, c – predicted by the classifier class.

The value of the percentage of correct answers for choosing a classifier must meet the condition presented in the equation (6):

$$Accuracy_{total} \to 1 \quad (6)$$

To classify the PRSs, we propose to use an algorithm based on a sub-count of the number of binary subsequences of length N-1 bits in the studied PRS. In [30], [31], it is noted that for example, for the sequence s = 1011010001, the frequency of occurrence of subsequences of length $N = 3$ bits is represented in Table II.

TABLE II. SUBSEQUENCES FREQUENCIES COUNTING

Subsequences	Number	Frequency
000	1	0.125
001	1	0.125
010	1	0.125
011	1	0.125

To restore the distribution of binary sequences, it is sufficient to analyze half of all possible subsequences. Thus, the dimension of the feature space for subsequences of length N bits is defined by the equation (7):

$$|S| = 2^{N-1} \quad (7)$$

To carry out the experiment, a sample of PRS was formed, consisting of 16000 files of 4 classes obtained as a result of file transformations containing meaningful text in Russian:

- Encrypted by algorithms AES, 3DES, RC4, and Camellia in CBC mode [32] – 4000 files.
- Archives RAR [33] – 4000 files.
- Archives ZIP [33] – 4000 files.
- Archives 7Z [33] – 4000 files.

The experiment was conducted in a software environment Anaconda [34].

Since the obtained values of the frequency of occurrence of sequences of length N bits are quite small values $(\sim 10^{-5}...10^{-6})$, the transition to a logarithmic scale of values was made to improve the accuracy of classification (logarithmic values).

Machine learning algorithms were used to construct classifiers and evaluate them [35]: a decision tree classifier (DTC), a logarithmic decision tree classifier (DTCL), a random forest classifier (RFC), and a logarithmic random forest classifier (RFCL). The obtained values of the accuracy of the PRS classification from the length of the subsequence N are shown in Fig. 3.

The obtained results indicate that it is possible to classify PRSs generated by encryption, compression algorithms, and pseudo-random number generators using the proposed algorithm with an accuracy greater than 0.95 for a 9-bit sequence length.

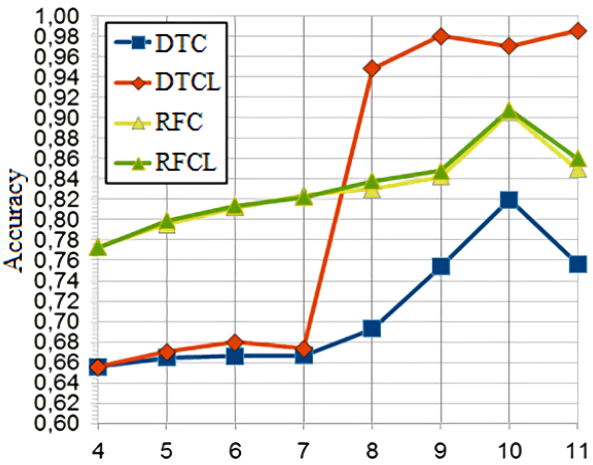

Fig 3. Accuracy for classification 4 classes of PSR.

During the experiments, 2 algorithms for constructing classifiers were used: the algorithm for constructing a decision tree and the algorithm for constructing a random forest. The algorithm for constructing the decision tree showed a higher accuracy of the PRS classification. To improve the accuracy of the classifier, the values of the frequency of occurrence of subsequences were converted to the logarithmic scale, which made it possible to achieve the accuracy of the PRS classification of 0.98.

IV. Conclusion

Modern DLP systems are not able to detect encrypted or compressed data with high accuracy, which allows you to use the data transmission channel in encrypted or compressed form, if there is no information about the compression algorithm, for transmitting confidential data. In this paper, we proposed a classification algorithm consisting of several stages: determining the most significant statistical features of random sequences on a training sample of data using the random forest algorithm and directly classifying the algorithm for building a decision tree. The proposed algorithm for feature extraction and classification allowed us to increase the accuracy of classification of encrypted and compressed data to an accuracy of 0.98.

Acknowledgment

The reported study was funded by Russian Ministry of Science (information security, project number 18/2020).

References

[1] Data Breach Report: A Study on Global Data Leaks in H1 2018, InfoWatch, https://www.infowatch.ru/analytics/reports. (Access date 14.01.2020).

[2] B.B. Mahesh, M.S. Bhanu, "Prevention of insider attacks by integrating behavior analysis with risk based access control model to protect cloud", Procedia Computer Science, Vol. 54, 2015, pp. 157-166.

[3] D. Kolevski, K. Michael, Cloud computing data breaches a socio-technical review of literature, 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), Greater Noida, India, 2015, pp. 1486-1495.

[4] S. Alneyadi, E. Sithirasenan, V. Muthukkumarasamy, Detecting Data Semantic: A Data Leakage Prevention Approach, IEEE Trustcom/BigDataSE/ISPA, Helsinki, Finland, Vol. 1, 2015, pp. 910-917.

[5] S. Alneyadi, E. Sithirasenan, V. Muthukkumarasamy, Discovery of potential data leaks in email communications, 10th International Conference on Signal Processing and Communication Systems (ICSPCS), Gold Coast, Australia, 2016, pp. 1-10.

[6] X. Huang, Y. Lu, D. Li, M. Ma, A novel mechanism for fast detection of transformed data leakage, IEEE Access, Vol. 6, 2018, pp. 35926-35936.

[7] K. Kaur, I. Gupta, A. K. Singh, Comparative Evaluation of Data Leakage/Loss prevention Systems (DLPS), In Proc. 4th Int. Conf. Computer Science & Information Technology (CS & IT-CSCP), 2017, pp. 87-95.

[8] L. Cheng, F. Liu, D. Yao, Enterprise data breach: causes, challenges, prevention, and future directions, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, Vol. 7, No. 5, 2017, pp. 1211.

[9] X. Shu, D. Yao, E. Bertino, Privacy-Preserving Detection of Sensitive Data Exposure, IEEE Transactions on Information Forensics and Security, Vol. 10, No. 5, 2015, pp. 1092-1103.

[10] F. Liu, X. Shu, D. Yao, A. R. Butt, Privacy-preserving scanning of big content for sensitive data exposure with MapReduce, Proceedings of the 5th ACM Conference on Data and Application Security and Privacy, 2015, pp. 195-206.

[11] X. Shu, J. Zhang, D. Yao, W. Feng, Rapid and parallel content screening for detecting transformed data exposure, Proceedings of the Third International Workshop on Security and Privacy in Big Data, 2015, pp. 191-196.

[12] Shu X., Zhang J., Yao D. D., Feng, W. C., Fast Detection of Transformed Data Leaks, IEEE Transactions on Information Forensics and Security, Vol. 11, No 3, 2016, pp. 528-542.

[13] Yu, X., Tian, Z., Qiu, J., & Jiang, F. , A data leakage prevention method based on the reduction of confidential and context terms for smart mobile devices, Wireless Communications and Mobile Computing, 2018. DOI: 10.1155/2018/5823439.

[14] X. Shu, D. Yao, E. Bertino, Privacy-Preserving Detection of Sensitive Data Exposure, IEEE Transactions on Information Forensics and Security, Vol. 10, No. 5, 2015, pp. 1092-1103.

[15] Shvartzshnaider Y., Pavlinovic Z., Balashankar A., Wies T., Subramanian L., Nissenbaum H., Mittal P., VACCINE: Using Contextual Integrity For Data Leakage Detection, The World Wide Web Conference, 2019, pp. 1702-1712.

[16] Kavitha T., Rajitha O., Thejaswi K., Muppalaneni N. B. Classification of encryption algorithms based on ciphertext using pattern recognition techniques, International conference on Computer Networks, Big data and IoT, 2018, pp. 540-545.

[17] C. Tan, Q. Ji, An approach to identifying cryptographic algorithm from ciphertext, 8th IEEE International Conference on Communication Software and Networks, 2016, pp. 19-23.

[18] C. Tan, Y. Li, S. Yao, A Novel Identification Approach to Encryption Mode of Block Cipher, 4th International Conference on Sensors, Mechatronics and Automation, Zhuhai, China, 2016. DOI: 10.2991/icsma-16.2016.101.

[19] C. Tan, X. Deng, L. Zhang, Identification of Block Ciphers under CBC Mode, Procedia Computer Science, Vol. 131, 2018, pp. 65-71.

[20] Ray P. K., Ojha S., Roy B. K., Basu A., Classification of Encryption Algorithms using Fisher's Discriminant Analysis, Defence Science Journal, Vol. 67, No. 1, 2017, pp. 59-65.

[21] Pan J., Encryption scheme classification: a deep learning approach, International Journal of Electronic Security and Digital Forensics, Vol. 9, No. 4, 2017, pp. 381-395.

[22] Wang, W., Zhu, M., Zeng, X., Ye, X., & Sheng, Y., Malware traffic classification using convolutional neural network for representation learning, International Conference on Information Networking (ICOIN), 2017, pp. 712-717.

[23] Wang W., Zhu M., Wang J., Zeng X., Yang Z., End-to-end encrypted traffic classification with one-dimensional convolution neural networks, IEEE International Conference on Intelligence and Security Informatics (ISI), 2017, pp. 43-48.

[24] Lotfollahi M., Siavoshani M. J., Zade R. S. H., Saberian M., Deep packet: A novel approach for encrypted traffic classification using deep learning, Soft Computing, 2017, pp. 1-14.

[25] Zhang J., Chen X., Xiang Y., Zhou W., Wu J. Robust network traffic classification, IEEE/ACM Transactions on Networking, Vol. 23, No. 4 , 2015, pp. 1257-1270.

[26] Pacheco F., Exposito E., Gineste M., Baudoin C., Aguilar J., Towards the deployment of machine learning solutions in network traffic classification: a systematic survey, IEEE Communications Surveys & Tutorials, Vol. 21, No. 2, 2018, pp. 1988-2014.

[27] Hahn D., Apthorpe N., Feamster N., Detecting compressed cleartext traffic from consumer internet of things devices, arXiv preprint arXiv:1805.02722, 2018.

[28] Casino F., Choo K. K. R., Patsakis C., HEDGE: efficient traffic classification of encrypted and compressed packets, IEEE Transactions on Information Forensics and Security, Vol. 14, No. 11, 2019, pp. 2916-2926.

[29] Tang Z., Zeng X. and Sheng Y., Entropy-based feature extraction algorithm for encrypted and non-encrypted compressed traffic classification International Journal of ICIC, Vol. 15, No 3, 2019.

[30] Khakpour A. R., Liu A. X., An information-theoretical approach to high-speed flow nature identification, IEEE/ACM transactions on networking, Vol. 21, No. 4, 2012, pp. 1076-1089.

[31] Konyshev M. U., Dvilyansky A. A., Barabashov A. Y., Petrov K. Y., Formation of probability distributions of binary vectors of the error source of a Markov discrete memory link using the method of "grouping probabilities" of error vectors, Industrial ACS and controllers, No. 3, 2018, p. 42.

[32] Konyshev M. U., Dvilyansky A. A., Petrov K. Y., Ermishin G. A., Algorithm for compression of a distribution series of binary multidimensional random variables, Industrial ACS and controllers, No. 8, 2016, pp. 47-50.

[33] Toolkit for the transport layer security and secure sockets layer protocols, http://openssl.org (Access date: 14.01.2020).

[34] Archive manager WinRAR, http://rarlab.com, (Access date: 14.01.2020).

[35] Programm environment Anaconda, https://www.anaconda.com/distribution/, (Access date: 14.01.2020).

[36] Breiman L., Classification and regression trees, Routledge, 2017, p. 358.

ABOUT THE AUTHOR

Alexander Kozachok

Workplace: Academy of the Federal Guard Service of Russian Federation

Email: alex.totrin@gmail.com

Education: Received his PhD degree in Engineering Sciences in Academy of Federal Guard Service of the Russian Federation in December 2012; received his doctorate in Engineering Science in 2019.

Recent research direction: information security, unauthorized access protection, mathematical cryptography, theoretical problems of computer science.

Andrey Spirin

Workplace: Academy of the Federal Guard Service of Russian Federation

Email: spirin_aa@bk.ru

Education: Postgraduate student in Academy of the Federal Guard Service of Russian Federation.

Recent research direction: information security, DLP systems, machine learning algorithms, classification of binary sequences.