Kryptografická metóda ochrany údajov pomocou generovania kľúčov s náhodnou dĺžkou [1]

Serhii OSTAPOV[*] – Liliia SHUMYLIAK[**] – Luboš CIBÁK[***] – Bohdan SHYLOV[****]

A cryptographic method of data protecting using random length key generation

Abstract

This article is devoted to the problem of data protection and the implementation of such protection using an encryption algorithm based on the multiple (multiple) use of coding algorithms with the addition of a random key. The robustness of such an algorithm could make it popular in areas sensitive to data privacy, such as banking.

Keywords: *encryption, data security, privacy, cryptographic system.*

JEL Classification: O30, P11, M15

Introduction

In economics and management, data encryption is used to protect confidential information and ensure the security of operations.

Banks, financial institutions, and financial companies use encryption to protect financial transactions, personal customer data, and confidential company information. In business, encryption is used to protect confidential data about products, production processes, costs, profits, and other confidential information. Companies use encryption to protect sensitive information about new technologies, innovations and intellectual property from unauthorized access and theft. Companies in retail, telecommunications, marketing, etc. use encryption to protect their customers' personal data, such as names, addresses, credit card numbers, etc. Data encryption is used to comply with data protection laws such as GDPR in the European Union or HIPAA in the United States, and to reduce the risk of data leakage and fines for data security breaches. The use of encryption can ensure the security of data throughout the supply chain, including customers, suppliers, logistics and other parties.

In all of these cases, encryption helps protect confidential information, maintain the trust of customers and partners, and comply with legal requirements.

Therefore, data encryption plays an important role in ensuring data security and protecting confidential information, which is critical to the successful functioning of a business and business operations.

The article considers the possibility of creating a new cipher that will be more reliable than existing ciphers, will have sufficient flexibility and will have other positive characteristics (such as information compression). The creation of such a cipher will provide an opportunity to better protect information and also reduce the size of coded data transmitted over the communication channel. To create such a cipher, it was decided to investigate the multiple uses of coding algorithms and add a key to such coding systems.

---

[1] The work was funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I03-03-V01-00085.

[*] Serhii Ostapov, D.Sc., Yuriy Fedkovych Chernivtsi National University, Software Department, Kotsyubynsky 2, Chernivtsi, 58012, Ukraine, e-mail: s.ostapov@chnu.edu.ua

[**] Liliia Shumyliak, PhD, Bratislava University of Economics and Management, Public Administration Institute, Department of Management Informatics, Furdekova 16, 851 04 Bratislava; Yuriy Fedkovych Chernivtsi National University, Software Department, Kotsyubynsky 2, Chernivtsi, 58012, Ukraine, e-mail: l.shumylyak@chnu.edu.ua

[***] doc. Ing. Luboš Cibák, PhD. MBA, Bratislava University of Economics and Management, Public Administration Institute, Department of Management Informatics, Furdekova 16, 851 04 Bratislava, e-mail: lubos.cibak@buem.sk

[****] Bohdan Shylov, Yuriy Fedkovych Chernivtsi National University, Software Department, Kotsyubynsky 2, Chernivtsi, 58012, Ukraine, e-mail: shylov.bohdan@chnu.edu.ua

Objectives of the research are the study of methods of data encryption in cryptography to ensure information security, risk management and solving new problems of protecting confidential data in the digital economy.

1 Literature review

Reliable encryption requires that the encrypted data have as uniform a distribution of symbols as possible, which means that statistical analysis cannot be used. Also important is the fundamental impossibility of obtaining the input text (the possibility of obtaining many logical decrypted messages from an encrypted message) [1].

To solve the first problem, it is proposed to use already existing coding algorithms. For example, Huffman's adaptive algorithm [2]. This algorithm involves determining a code word for each input word. The length of the code word and the input word encrypted with it are constantly changing. After several iterations of using this algorithm, the probability of symbols becomes almost the same. The problem with using Huffman's adaptive algorithm directly is that this algorithm does not provide a key. And in order to correctly read a coded message, you need a table of code words and knowledge of the beginning of the next code word.

There are other improvements to the classic Huffman algorithm [3], such as arithmetic coding [4]. The problem of using such algorithms is the need to re-pass the data. First, the frequencies are read and the table is formed, and during the second pass, the actual coding takes place. Algorithms with a double pass and determining the frequency of occurrence of symbols can be optimal, while algorithms using a single pass are usually inefficient [5]. Despite the inefficiency of coding with one-pass algorithms, they can be easily implemented in the form of parallel computing if several iterations of coding take place. Also, single-pass algorithms usually involve changing the codeword, which leads to a more chaotic distribution of data and more difficult and sometimes impossible statistical analysis.

2 Research methodology

The research methodology described in the article revolves around the development and testing of a cryptographic method for data protection using random length key generation. Generating and adding a key using a specific algorithm is proposed. This algorithm includes steps for adding leading zeros and additional parameters to the key to enhance decryption complexity.

The methodology is illustrated with figures showing the encryption and decryption algorithms (Figures 1 and 2) and a step-by-step encryption and decryption process (Figure 3). The robustness of the proposed algorithm is tested using the NIST STS statistical test package, which includes 188 tests.

The results are presented in a table and a figure, indicating the number of tests passed across different iterations. The tests measure the uniformity and entropy of the encrypted data, crucial for evaluating the security of the cryptographic method.

The article discusses the complexity of decrypting data without the known key, emphasizing the high computational power required for cracking the encryption. The complexity is calculated based on the number of possible combinations for selecting code word tables, considering the length of the encoded words and the limit for overflow.

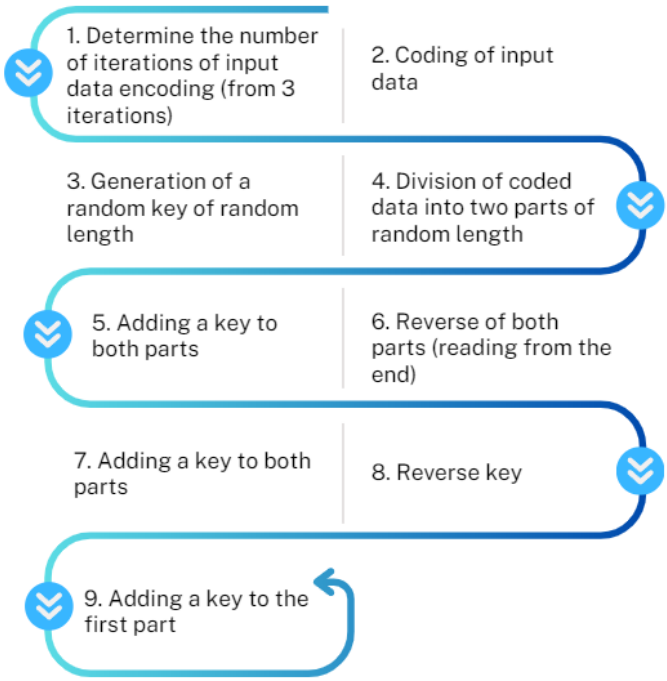
The research includes graphical representations (Figure 4) to depict key placements and the results of encryption and decryption processes. These visual aids help in understanding the flow and security aspects of the proposed method.

Overall, the methodology combines theoretical development, statistical testing, and practical implementation to validate the effectiveness of the proposed cryptographic method for data protection.

3 Proposed method

It is proposed to generate and add a key using the following algorithm:

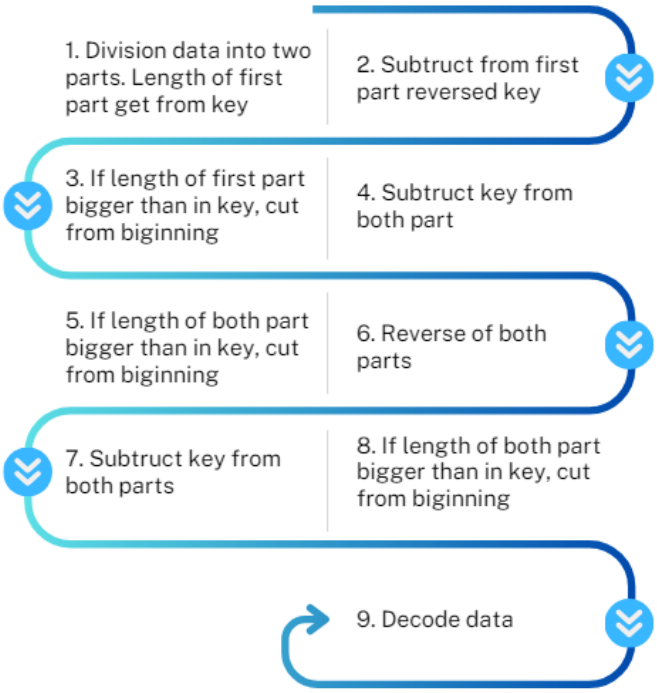
Figure 1 Encryption algorithm



Source: created by authors

An additional element of the complexity of decryption in case adding data with leading zeros, it is not always clear what the data was before the key was added. For example, adding the data 00101 to the key 01001, we will get encrypted data 01110.

Figure 2 Decryption algorithm



Source: created by authors

But in the reverse operation, subtracting the key from the encrypted data, 00101, 0101 and 101 may be correct, which is critical for correct decryption. To solve this problem, it is proposed to add additional parameters to the key containing length of the data before each addition operation performed during the encryption steps. The key in this case will look like this:

- Number of iterations
- The length of the first part of data
- The length of first part before adding key
- The length of second part before adding key
- The length of first part after adding key
- The length of second part after adding key
- The length of first part after adding key second time
- Key

An example of encryption and decryption for random data and a key with one encoding iteration is shown in Figure 3.

For crack, you need to find a key that can be of any length, and therefore unbreakable for modern computing power. Another option for crack is to select a table of code words for the encoding algorithm and use it for a part of the text that is not supposed to be protected by the key (if the length of the key is much shorter than the length of the data). The difficulty of such selection is that there are many variants for the initial configuration of the table. The number of variants depends on the length of the word to be coded and the limit on the number of occurrences for overflow (usually these are integer types in programming languages, which in practice means more than 2,000,000,000). The number of variants of the codeword table can be calculated as (limit for overflow) ^ (number of variants of codewords). If the length of the encoded word is 8 bits, the number of encoded words will be 28, which is 256. Then the complexity of the table selection will be (2 000 000 000)256, which is equal to 1.1579E2381. This selection complexity will be for one coding iteration. For correct decoding, you need to select the tables for all iterations. So, for 10 iterations of coding, 1.1579E23810 combinations need to try for correct decoding.

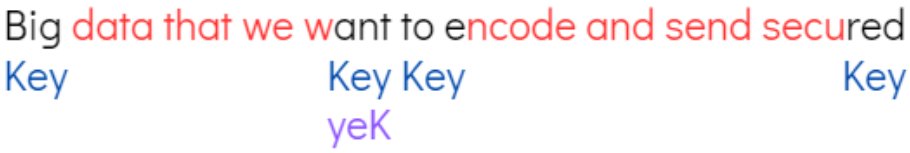
Figure 3 Step-by-step encryption and decryption

```
Data: 01111011001
Key: 0111100001011
Iteration: 3
After coding: 0000100000000110101111000010000000000110001100001100000000000000110000
First part length: 40
First part:0000100000000110101111000010000000000110
Second part:0011000011000000000000000110000
First part after add key:0000100000000110101111000010111100010001
Second part after add key:0011000011000000000111100111011
First part reversed:1000100011110100001111010110000000010000
Second part reversed:1101110011110000000001100001100
First part after add key second time:1000100011110100001111010110111100011011
Second part after add key second time:1101110011110000000100000010111
First part after add reversed key:1000100011110100001111011000100100111001
Encode: 1000100011110100001111011000100100111001110111001111000000100100001011 1
Key to decode: 0000001100000000000101000000000000010100000000000000111110000000000101000000000000001111100000000001010000111100001011
First part after separating:1000100011110100001111011000100100111001
Second part after separating:1101110011110000000100000010111
First part after subtract reverse key:1000100011110100001111010110111100011011
First part subtract key first time:1000100011110100001111010110000000010000
Second part subtract key first time:1101110011110000000001100001100
First part reversed:0000100000000110101111000010111100010001
Second part reversed:0011000011000000000111100111011
First part subtract key second time:0000100000000110101111000010000000000110
Second part subtract key second time:0011000011000000000000000110000
Decode: 01111011001
```

Source: created by authors

In Figure 4, the key is shown in blue, the key after reverse is shown in purple, the data protected by the key is in black, and the data that can be cracked using the selection of the coding table is shown in red.

Figure 4 Graphic representation of key placement



Source: created by authors

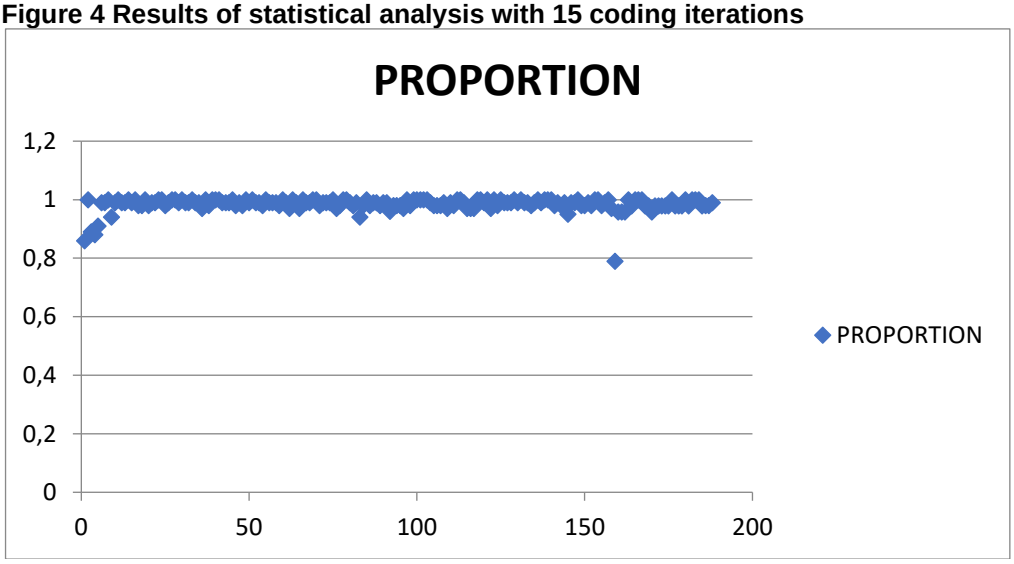
4 Results and discussion

Statistical analysis was performed for this algorithm using the NIST STS statistical test package (188 tests). The test results are shown in the table and figure below.

Table 1

Number of iterations	3	10	15
Number of passed tests	10	176	176
Number of passed tests with result between 0.96 and 0.97	3	6	1
Number of passed tests with result between 0.97 and 0.98	2	14	24
Number of passed tests with result between 0.98 and 0.99	2	33	33
Number of passed tests with result between 0.99 and 1	3	60	57
Number of passed tests with result 1	0	63	61
Average value of tests result	0.579567901	0.975101064	0.984675532

Figure 4 Results of statistical analysis with 15 coding iterations



Source: created by authors

As can be seen from the results of the statistical analysis, most of the tests passed with a good value and only a few tests did not pass. To fix this, the function of mixing the result can be added.

However, since the statistical analysis test the occurrence rate and entropy of the cipher and was designed to test encryption algorithms based on shifts and shuffling using the same key for different blocks (such as in this paper [6]), the analysis does not reflect the full complexity and uniformity of data distribution in the algorithm based on replacement and periodic change of code words.

Conclusions

The use of coding methods in combination with the generation and addition of a key to multiple times coded data can be a promising direction in cryptography. The difficulty of matching a table to decrypt data without a known key, a key of random length, the need to know other encryption parameters besides the key (such as the lengths of the parts before addition, the length of the split parts, and the number of iterations) add additional complexity to crack. An additional advantage of the algorithm is that, in the general case, for large enough volumes of data, coding algorithms will give a result of a shorter length, which will reduce the volume of data to be transmitted. The reliability of the algorithm in combination with the reduction of the amount of data after encryption, as well as the synchronicity of the algorithm (decryption using the same key used for encryption) can positively affect all areas of modern life where private, confidential or important information is present. And the use of a faster and more reliable method of data protection can have a positive effect on most spheres of the economy (some directly, for example, the banking sphere, some indirectly).

Literature

[1] *Cryptanalysis.* [online]. [cit. 2024-03-15]. Available online: https://en.wikipedia.org/wiki/cryptanalysis

[2] *Adaptive Huffman coding. [online]. [cit. 2024-03-15]. Available online:* https://en.wikipedia.org/wiki/Adaptive_Huffman_coding

[3] *Huffman coding.* [online]. [cit. 2024-03-15]. Available online: https://en.wikipedia.org/wiki/Huffman_coding

[4] *Arithmetic coding*. [online]. [cit. 2024-03-15]. Available online: https://habr.com/ru/articles/130531/

[5] ROMANYUK, M. I., SAVCHENKO Yu. G.: *Basics of information theory and coding*. Kyiv: KPI named after Igor Sikorskyi, 2019. 22 p.

[6] KORDOV, KRASIMIR: Text encryption algorithm for secure communication, *International Journal of Applied Mathematics*, 2021.