Stability analysis of binary transposition algorithms during attacks on text data

L S Kryzhevich, V M Dovgal
Kursk State University, Kursk, Radisheva str., 33, 305000, Russia
E-mail: leonid@programist.ru

Abstract. Different approaches to using of transposition methods for data protection are examined in the article. The author has conducted comparative analysis of the approaches to increasing strength to protect the text from various attacks when its length and the data amount given per one symbol are raised. Basic criteria of evaluation of the offered methods are marked as well. The increase of the computational complexity in the described approaches is calculated.

Keywords: binary transpositions, binary data distribution, security evaluation, computing costs.

1. Introduction
Nowadays, volumes of communicated information increase each year. Especially, there are more and more amounts of personal and confidential data. From the one hand, it is connected with great development of messengers and social networks, which are subject to Article 23, Paragraph 2 of the Constitution of the Russian Federation and, as a consequence, to Article 138 of the Criminal Code of Russia that guarantee protection of our private correspondence. From the other hand, there is Federal Law N 152 FL "On Personal Data" where we can see that disclosure of personal information of a citizen carries a punishment. Thus, data collection and analysis and storage systems in automated systems or web resources [1] must follow vital security conditions, and all the information contained in a file storage should be enciphered [2].

Classical methods of cryptographic transformation include substitution ciphers and transposition ciphers of both stand-alone symbols and entire blocks of an original text.

Traditional cryptographic methods where transposition is used imply movement of stand-alone symbols of a secret message (see Figure 1) or replacement of higher least significant bits in data packets. It has been proved that they have a higher level of cryptographic strength than ciphers of block substitution [3]. We can look at their classical scheme.
The number of possible transpositions can be calculated according to the following formula:

\[ \hat{P}(n) = \frac{n!}{k_1! k_2! \ldots k_l!} \]  

(1)

where

- \( n \) – the number of symbols of the text;
- \( l \) – the number of signs in the alphabet;
- \( k_i \) – the number of repetitions of the i-symbol of the alphabet;

A transposition process gives a great number of text modifications that have the same set of symbols. This process can be illustrated with the help of two examples[4]. We take a word that consists of 6 symbols (for instance, the word "приказ"). The number of combinations in this case is \( 6! = 720 \) because there are no repeated letters in the word. However, none of these combinations (excepting "приказ" itself) has a lexical meaning in the Russian language. The next example relates to the phrase "секретное сообщение". This phrase consists of 19 symbols, some of which are repeated. It is necessary to pay attention to the fact that transpositions of the same symbols give the same words. Therefore, the number of transpositions is not \( 19! = 1,2 \cdot 10^{17} \), but, according to the formula (1), it is \( \approx 2,1 \cdot 10^{14} \).

Thus, methods of transposition of stand-alone symbols of a text give plenty of senseless messages, which can be excluded according to the vocabulary. If the generated combinations represent words that have lexical meanings in the examined language, they can be sorted out according to the message context. If this is the case, it is easy to find out meanings of encrypted words and phrases [5].

The following contradiction appears from the above-stated facts: from the one hand, it is necessary to get a cryptographic method where symbols of an original text are evenly distributed and the number of combinations is an order of magnitude greater than transpositions of stand-alone symbols; from the other hand, it must give all possible variants of words, which will not be excluded according to vocabulary. Thus, the objective of our research, which is determined by the necessity of developing new approaches to solving the cryptographic problem, is rather topical.

In order to solve this problem, we should analyze such an approach where the plaintext message is transposition during its transmission with the help of a communication pathway. In such a case, it is necessary to pay attention to the fact that bit swapping is applied not to stand-alone blocks of the file, but to the whole file to avoid use of several sets of keys and generation of periodic repetition [6].
Thus, when a binary code is used, the length of the message increases eightfold if ASCII is applied or it increases 16 times if Unicode is applied. It is important to notice that the number of symbols in the alphabet drops to 2 (0 and 1). Later it leads to increase of repetitions of the alphabet signs themselves.

According to the formula (1), the number of combinations of the cryptotext, which has been received with the help of transposition of bits, can be calculated in the following way:

\[
\hat{P}_s(n) = \frac{(8n)!}{k_0! k_1!}, \text{(ASCII)}
\]

or

\[
\hat{P}_u(n) = \frac{(16n)!}{k_0! k_1!}, \text{(Unicode)}
\]

where

- \(n\) – the number of the text symbols;
- \(k_0\) – the number of "0" symbols in the message code;
- \(k_1\) – the number of "1" symbols in the message code;

We can use the described approach to the above-stated examples and compare the results.

The phrase "секретное сообщение" from our first example has 152 binary bits in binary ASCII. In that case, the number of "0" symbols is \(k_0=81\), and the number of "1" symbols is \(k_1=71\).

If this occurs, the number of the ciphertext combinations according to the formula (2) can be calculated in the following way:

\[
\hat{P}_s(n) = \frac{(8 \cdot 19)!}{71! \cdot 81!} = 2.7 \cdot 10^{44}.
\]

This result \(\hat{P}_s(n) = 2.7 \cdot 10^{44}\) is 30 orders of magnitude greater than the result \(\hat{P}(n) = 2.1 \cdot 10^{14}\), which is received with the help of the transposition of the message symbols.

If a sixteen bit encoding is used, then the number of cryptotexts according to the formula (3) increases significantly:

\[
\hat{P}_u(n) = \frac{(16 \cdot 6)!}{84! \cdot 220!} = 3.8 \cdot 10^{76}.
\]

Thus, doubling the message capacity makes the number of combinations 32 orders of magnitude greater in comparison with \(\hat{P}_s(n)\).

We can find out how much the number of transpositions decreases if the bit count swerves from the central value.

The maximum value of transpositions can be reached when the number of "0" symbols and the number of "1" symbols are equal. If we take into account our example where an eight bit encoding is used, these values are \(k_0=76\) and \(k_1=76\). It means that the number of transpositions can be calculated in the following way:

\[
\hat{P}_s(n) = \frac{(8 \cdot 19)!}{76! \cdot 76!} = \frac{152!}{76! \cdot 76!} = 2^{76} \cdot 152!! = 3.8 \cdot 10^{44}.
\]

We can pay attention to the fact that the departure of the correlation between the number of "0" symbols and the number of "1" symbols from the central value leads to a slight decrease of the number of the transposition combinations, which is less than one order of magnitude.
If a sixteen bit encoding is used, the result is formulated in the following way:

\[
\hat{P}_{16}(n) = \frac{(16 \cdot 19)!}{152! \cdot 152!} = \frac{304!}{152! \cdot 152!} = \frac{2^{152} \cdot 304}{152!} = 1.5 \cdot 10^{90}
\]

The difference is more significant here. It is about 14 orders of magnitude between \(\hat{P}_{16}(84 + 220) \approx 3.8 \cdot 10^{36}\) and \(\hat{P}_{16}(152 + 150) \approx 1.5 \cdot 10^{90}\).

We are going to examine an issue of changing the transpositions number when the symmetry of distribution deviates from the uniform value.

Bits can be grouped together in bytes at random. Therefore, any words of the used language that consist of the same set of bits can be formed. These combinations are not excluded according to the vocabulary. In that case, it is necessary to use an exhaustive search. Supercomputers are applied to sort out such great amounts of combinations.

There is a necessity to find a criterion of minimum computing costs. When this criterion is overcome with the help of the encryption algorithm, an eavesdropper will not be able to decode the message if he does not know the key.

At the moment, the most powerful supercomputer in the world is Summit (the USA) [7]. (Its productive capacities are 200 PFLOPS = \(200 \cdot 10^{15}\) operations per second). Let us suppose that one operation is a check of one combination (according to the most optimistic forecast). The necessary time that is spent on sorting out the variants of the above-stated example can be calculated in the following way:

\[
T_{16}(n) = \frac{64,35 \cdot 10^{27}}{200 \cdot 10^{15}} = 3.22 \cdot 10^{10} \text{ sek}.
\]

We know that a year consists of 31,6 million seconds. Thus, it will take more than 1000 years to finish the described process.

The criterion "cryptographic strength over time" is the strength level of the algorithm where the time that is necessary to sort out the variants exceeds the age of the Universe (the age of the Universe is 13.7 ± 0.2 billion years; it is a bit less than \(1.4 \cdot 10^{10}\) years = \(4.42 \cdot 10^{17}\) seconds) [8].

The general computation power of all computing systems of the world is approaching to \(10^{20}\) FLOPS [9]. It imposes an additional requirement on the level of the cryptographic strength of the algorithm [10]. Thus, if all computers of the world are attacked with a "Trojan horse" virus and later are used to look over all the transposition combinations, it must take so much time as at least the current age of the Universe is. As a sequence, the number of transpositions that are made with the help of this cryptographic algorithm should not be less than \(4.42 \cdot 10^{37}\).

In order to solve this problem, it is possible to use two directions of the research. They are an increase of the text length and an increase of the length of binary representation of the letters [11]. We are going to examine these two directions separately.

2. An increase of the text length.

The purpose of the method is to find the minimum text length that is enough to clear the threshold of cryptographic strength over time.

We should compute the number of transpositions in its binary representation. It can be calculated with the help of the formula (4):
According to the above-stated facts, \( P(n) = \frac{(k_0 + k_1)!}{k_0! k_1!} > 4.42 \cdot 10^{37} \) (4)

The covered dependence (4) is portrayed in Figure 2.

When the text length \( n \) is fixed, the maximum value of the transpositions can be reached if \( k_0 = k_1 \).

It is necessary to get the lower bound of the number of \( n \) symbols in the original message. Thus, we have:

\[
k_0 = k_1 \Rightarrow k_0 + k_1 = 2k
\]

(5)

It means that the condition (4) can be written in the following way:

\[
\max \left( \frac{(k_0 + k_1)!}{k_0! k_1!} \right) = \frac{(k + k)!}{k! k!} = \frac{(2k)!}{(k!)^2} > 4.42 \cdot 10^{37}
\]

We should get limitations on the number \( n_{\text{min}} = \frac{k}{4} \) from the last condition and the condition (5):

\[
\frac{(2k)!}{(k!)^2} = 2^k \cdot \frac{(2k-1)!!}{k!} = 2^k \cdot \prod_{i=1}^{k} \frac{2 \cdot i - 1}{i},
\]

After evaluation of the resulting expression, we have:
\[
\frac{2^{2k-1}}{k} < 2^k \cdot \prod_{i=1}^{k} \frac{2 \cdot i - 1}{i} < 2^{2k}
\]

According to the fact that \(2^{125} < 4.42 \cdot 10^{37} < 2^{126}\), we receive the limitation \(k \geq 65\).

Taking into account the condition about an integrality of the length of the bit sequence \(n\), we can see that the threshold of "cryptographic strength over time" can be cleared when the length of the original message is more than 16 symbols if an eight bit encoding is used. If a sixteen bit encoding is used, the length of the original message should be more than 8 symbols respectively. It lets us use this method to keep even short passwords.

The next stage should be dedicated to the analysis of influence of the text length enlargement on the amount of time that is spent to look over the text [12]. We are going to find out how much the number of operations increases if the text is increased by 1 bit. Let us say that the number of “0” symbols has grown in the text encoding to raise certainty. In such a case, we have:

\[
P_L = \frac{P(n + 1)}{P(n)} = \frac{(n + 1)!}{(k_0 + k_1 + 1)!} \cdot \frac{k_0! \cdot k_1!}{(k_0 + 1)! \cdot k_1!} = \frac{k_0 + k_1 + 1}{k_0 + k_1} = 1 + \frac{k_1}{k_0 + 1}.
\]  

(6)

Figure 3. Comparison of the histograms of transpositions for 128 bits and 129 bits
Thus, we can see that the complexity increases in 
\[ L = 1 + \frac{k_1}{k_0 + 1} \]
and depends on the asymmetry level of distribution of binary "0" and "1" in the text encoding and \( L \in [1; n+1] \).

We should pay attention to the fact that the \( L \) index that is used in the formula (6) does not depend on a type of the chosen text encoding, but it depends on the deviation of \( k_0 \) and \( k_1 \) from the central value. This regularity is portrayed in Figure 4.

3. An increase of the length of the symbol encoding.

It is necessary to analyze the second issue, the purpose of which is to find out how the capacity increase influences the rise of complexity of computations [13].

Let us say that \( n \) – the number of the text symbols,
\[ k_0 \] – the number of "0" symbols in the message code;
\[ k_1 \] – the number of "1" symbols in the message code.

If this is the case, we can use the formula (2) to calculate the number of transpositions in binary representation of the text, the length of which is \( n \) symbols and it is expressed with the help of an eight bit encoding. It is, of course, meant in the formula (2) that \( k_0 + k_1 = 8n \).

From the other hand, we can transform the formula (3) in order to calculate the number of transpositions in the same text, which is expressed with the help of a sixteen bit encoding:

\[ P_{16}(n) = \frac{(16n)!}{(K_0)!(K_1)!} \approx \frac{(2 \cdot 8 \cdot n)!}{(2k_0)!(2k_1)!}, \]

where \( K_0 \cdot K_1 = 16n \).

Figure 4. Comparison between the histograms of different transpositions for 129 bits
In order to find out how much the computing costs increase when the length of the text encoding is enlarged, we can divide the formula (2) by the formula (7):

\[
\frac{P_{16}(n)}{P_8(n)} = \frac{(2 \cdot 8 \cdot n)!}{(2k_0)!(2k_1)!} \cdot \frac{(8 \cdot n)!}{(k_0)!(k_1)!} = \frac{(16n-1)!}{(2k_0-1)!(2k_1)!} \cdot 2^{2k_1} \Rightarrow \frac{P_{16}(n)}{P_8(n)} = \frac{(16n-1)!}{(2k_0-1)!(2k_1-1)!};
\]

Thus, we have received the final formula that can be used to calculate the increase of the complexity of computations in order to look over the encoding combinations when a conversion key is changed from 8 bits to 16 bits (for example, when we face a switch from ASCII to Unicode).

This article has dealt with the analysis of different aspects of use of the binary transposition approach depending on the encoding method (8) of the text and its length. It has been shown that this approach produces several orders of magnitude of bit combinations more than traditional transposition methods do[14]. However, the method is especially effective when the text length is bigger than 65 bits. We have developed the criterion (6) that determines possible ways of increasing the cryptographic strength of the algorithm by adding the definite bits into the text. The prospects and advantages of this approach are described [15]. One of them is reducing possibilities of sorting out the deciphered parts of the text according to the context or vocabulary if words of a natural language are used. Another important aspect is impossibility to find a bit while encryption access passwords or keys that are generated with the use of pseudorandom sequences, even if an eavesdropper has a part of the plain text[16].

References

[1] Loik R 2016 Integration Trends of EU Internal Security and Law Enforcement: How Legal, Technological and Operational Advancements Matter Baltic Journal of European Studies 6(2) pp 3-27.
[2] Ho Ngoc Duy 2012 Algorithms of information processing in automated systems of an electronic document flow Thesis for a Candidate Degree in Technical Sciences p161.
[3] Singh S and Kleist V 2005 The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography IEEE Annals of the History of Computing 24(2) pp 97-98.
[4] Shaska T and Ustimenkob V 2009 On the homogeneous algebraic graphs of large girthand their applications Linear Algebra and its Applications pp 1826-1837.
[5] Fix A S 2014 The history of cryptography and new ethic status of information Online scientific journal “Philosophical problems of IT and Cyberspace” (Volume 2 Issue 8) pp 65-76.
[6] Kant S, Verma N 2000 An effective source recognition algorithm: extraction of significant binary words Pattern Recognition Letters 21(11) pp 981-988.
[7] Abramov S M 2019 June 2019: Analysis of the Development of the Supercomputer Industry in Russia and Worldwide Program Systems: Theory and Applications 10(3) p 3-40
[8] Avelino A and Kirshner R 2016 the dimensionless age of the universe: a riddle for our time *The Astrophysical Journal* 828(1) 35

[9] Levin V K 2014 Supercomputer trends *Computational nanotechnology: The scientific journal concerning: Mathematics, Physics, Economics and economic sciences, Automation. Computer engineering* (Publishing house Yur-VAK) pp 35-38

[10] Moldovyan A A 2005 Synthesis methodology of algorithms of information security in computer systems based on controlled substitution-permutation networks *Abstract of a thesis for a Doctoral Degree in Technical Sciences* (Saint Petersburg: Printing office VAS) p 44.

[11] Laskari E C, Meletiouc G C, Stamatiou Y C, Vrahatis M N 2005 Evolutionary computation based cryptanalysis: A first study *Nonlinear Analysis: Theory, Methods & Applications* 63(5-7) pp 823-830.

[12] Doshina A D, Mikhaylova A E , Karlova V V 2015 Cryptography. Basic methods and problems. Modern cryptography trends *Modern trends of technical sciences: proceedings of the IV International scientific conference* (Kazan: Buk ) pp 10-13

[13] Chernyshev Yu O, Dubrov E O, Ryazanov A N, Sergeev A S 2013 Development of methods of cryptanalysis of transposition and substitution ciphers in information security systems on the basis of evolutionary and optimization methods *Reds: telecommunication devices and systems* Volume: 3 Issue: 3 (Moscow: Publisher: Russian scientific and technical society of radio engineering, electronics and communication of name A.S. Popov) pp 221-225.

[14] Laskari E C, Meletiou G C, Stamatiou Y C , Vrahatis M N 2007 Applying evolutionary computation methods for the cryptanalysis of Feistel ciphers *Applied Mathematics and Computation* 184 (1 SPEC. ISS.) pp 63-72

[15] Sotov L S 2011 The hardware implementation of permutation instructions of bits of the data *Heteromagnetic microelectronics* Issue: 10 (Saratov: OAO Research institute "Tantal") pp 25-50.

[16] Chávez M L 2005 Security services in fieldbuses: at what cost? *IFAC Proceedings Volumes (Electronic Materials* Volume 38 Issue 2) pp 166-171