Информатика и вычислительная техника

UDC 004.056.55

Symmetric Encryption on the Base of Splitting Method

V. L. Stefanyuk[*][†], **A. H. Alhussain**[†]

[*] *Institute of Information Transmission Problems, Moscow, Russia*

[†] *Peoples' Friendship University of Russia, Moscow, Russia*

This article shows a method of secured transmitting of information by using splitting encryption algorithm which replaces each character in plaintext by k-integer in ciphertext. Splitting algorithm is a generalization of the secured transmission procedure with secret key that.

This study shows how to use a set of cryptographic keys which are generated using genetic algorithm and pseudorandom number generators, to solve some of serious problems in the modern cryptography.

Key words and phrases: genetic algorithm, pseudorandom number generator, encryption, decryption, cryptography, monomorphism, splitting algorithm

1. Introduction

The problems of information protection are excited the humanity for centuries. The need of information security has originated from the necessary of diplomatic negotiation, secret transferring of the military information, and protection of the personal information.

In the recent years, the information has become considered as financial category, this add more attraction and attention to the data security. Protection of the text during transmission via communication channels is an important task for business applications, and many other areas of the modern life [1].

There are several encryption algorithms; one of them is XOR encryption which uses pseudorandom number generator (PRNG). The experience has shown that XOR encryption has relative weakness against the actions of experienced hackers, and it is not entirely satisfy the requirements of high level of security [1, 2].

The principle of XOR encryption could be summarized as follows: generate keystream using pseudorandom numbers generators after that apply XOR operation (modulo-2 addition) between the obtained cryptographic keys and plaintext.

Modulo-2 addition in XOR encryption can be accomplished in several ways, for example, by the formula [1]:

$$y = x \oplus k, \tag{1}$$

where y – ciphertext, x – ASCII code plaintext, k – the generated cryptographic keys using PRNG; and $\oplus$ — bitwise "exclusive or". The schema of XOR cipher is shown in fig. 1.

This article is proposed a symmetric encryption algorithm, which improves the safety of traditional encryption algorithms by replacing each character in plaintext by k-chain in ciphertext. This mechanism would increase the level of security and it has not been studied previously in cryptography.

This paper is organized as follows: in section 1 we define the proposed encryption algorithm based on the splitting method, in section 2 provides a description of splitting algorithm, in section 3 proves the property of monomorphism of splitting method. In section4 presents the results of experiments. The conclusion is contained in the final section.

Received 25th May, 2016.

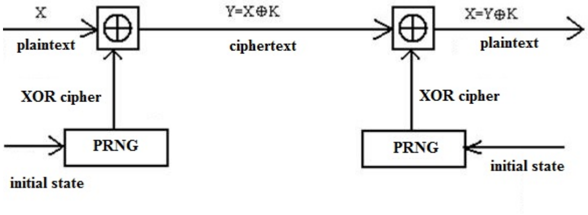


Figure 1. The scheme of XOR encryption

2. Defintion of the Encryption Algorithm Based on Sippliting Method

The term of splitting, which is referred in this paper, means replacing each character in plaintext by k-chain of integers in ciphertext; to be transmitted over a communication channel. Splitting provides defense in depth for the transmitted information from malicious actions of various kinds.

Definition: splitting k-level means representation of each character in plaintext as a sequence of k-integers in ciphertext.

The obtained ciphertext by this method is difficult to reveal, as the cryptographic keys are variable and the cipher changes randomly for each ciphered letter. This concept is new in cryptography, and there is no similar proposal has been issued or reported before.

In particular, the splitting algorithm provides reliable protection from cryptanalytic attacks based on counting the frequency of occurrence of the letters in the ciphertext. This algorithm does not depend on the probability distribution of the letters in the language or the other properties of the natural language. (If the primary requirement is the speed and size, it is possible to use k=1; But if the degree and level of security and privacy is more important, choose $k > 1$).

3. Description of Splitting Algorithm

3.1. Mathematical model of splitting algorithm

3.1.1. Mathematical model of the encryption algorithm

$$\text{encryption process} = \begin{cases} \text{splitting level} = 1 \rightarrow \text{modification XOR encryption} \\ \text{splitting level}\, k, \text{where}\, k > 1 \rightarrow \text{Encryption algorithm based} \\ \text{on siplliting method} \end{cases}$$

$$\text{encryption process}\, Y = \begin{cases} \text{KPRNG} \oplus \text{ASCII code, when}\, k = 1 \\ \text{quotient remainder}_k \ldots \text{remainder}_2\, \text{remainder}_1, \\ \text{when}\, k > 1\, \text{quotient} = \frac{\text{KPRNG}}{\text{ASCII code}} \\ \text{remainder} = \text{KPRNG mod ASCII code} \\ \text{KPRNG} > 256 \end{cases}$$

KPRNG denotes a generator that creates a sequence of conventional pseudorandom number generator (PRNG) after applying the operations of genetic algorithm (GA), which ensures a high probability of inability to predict the next character.

3.1.2. Mathematical model of decryption algorithm

$$\text{Decryption process} = \begin{cases} k = 1 \rightarrow \text{KPRNG} \oplus \text{ASCII code} \\ k > 1 \rightarrow \frac{\text{KPRNG}-\text{remainder}}{\text{quotient}} \end{cases}$$

3.2. The scheme of splitting algorithm

3.2.1. The scheme in case of $k = 1$

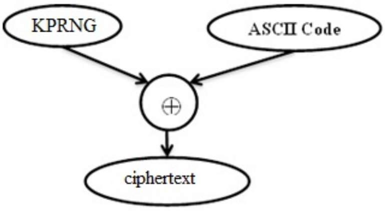


Figure 2. The scheme of splitting algorithm for $k = 1$

3.2.2. The scheme in case of $k > 1$

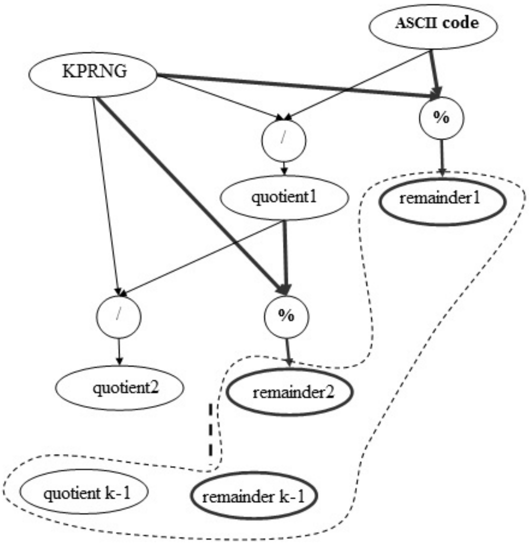


Figure 3. The scheme of splitting algorithm for k levels, $k > 1$

3.3. The secret key

The key – it is a particular secret state of some parameters of the cryptographic algorithm of the data that provides only one choice of all the possible options for the transformation encryption algorithm [1]. In the symmetric algorithm the same piece of information (i.e. key) is used to encrypt and decrypt the message [1].

The secret key in the proposed encryption algorithm contains information about the genetic algorithm, the level of splitting, and parameters of pseudorandom number generator. The block diagram of the secret key is shown in fig. 4.

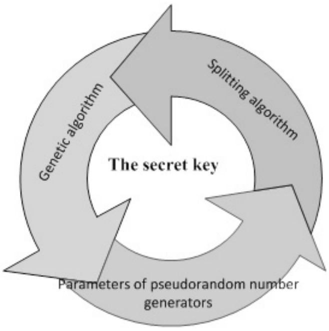


Figure 4. The block diagram of the secret key

The private key contains a set of parameters that make the cryptographic algorithm difficult for attacker to break it (increasing the level of security) [3]. These parameters are as follows:

1. The parameter of splitting algorithm, which indicates the "level of splitting". This parameter specifies the number of characters of ciphertext, which is replaced by substituting in place of each character in cleartext to be sent over the channel.
2. The parameters of genetic algorithm, which include "generation size, the number of generations, the length of the chromosome, the initial value, the end value" [4].
3. The parameters of pseudorandom number generator, which belong to the selected pseudorandom number generator [4]:
   (a) In the case of Blum-Blum-Shub and Fibonacci generators, parameters include "initial value and modulus".
   (b) In the case of a linear congruential generator, parameters include "initial value, modulus, increment and multiplier".
   (c) In the case of a quadratic congruential generator, parameters include "initial value, module, a, b, c".

3.4. Steps of the encryption algorithm

Input: cleartext, the type of generator, and secret key.

In case if the splitting level is 1

1. Generate a sequence of cryptographic keys, denoted this sequence by the symbol (S_0) , on the basis of the selected pseudorandom number generator, secret key, and the selected genetic algorithm.
2. Convert each character in cleartext into its ASCII code value. Let denoted this number by the symbol (S_{ASCII}).
3. Apply XOR operation between a part of the sequence (S_0), which is obtained in step1, and the ASCII code representation (S_{ASCII}), which is obtained in Step2, to obtain a new sequence, which we denote by the symbol C, where $C = S_0 \oplus S_{ASCII}$.
4. The sequence C is the ciphertext.

In the case of splitting level k, where $k > 1$

1. Generate a sequence of cryptographic keys, denoted this sequence by the symbol (S_0), on the basis of the selected pseudorandom number generator, secret key, and the selected genetic algorithm.
2. Convert each character in cleartext into its ASCII code value. Let denoted this number by the symbol (S_{ASCII}).

3. Select keys which have a value more than 256 from the generated sequence S_0.
4. For each character in cleartext apply division $k-1$ time and save the remainder of integer division at each step, after that save the final result and quotient.
5. The ciphertext will be a sequence of the form:

$$\text{quotient}_{1k}\text{remainder}_{11}\text{remainder}_{12}\ldots\text{remainder}_{1k},\ldots,\\ \text{quotient}_{nk}\text{remainder}_{n1}\text{remainder}_{n2}\ldots\text{remainder}_{nk}.$$

4. The Main Theorem of Splitting Algorithm

4.1. Definition of the mathematical function of the splitting algorithm

Suppose we have a character α_i, which is the ASCII code of the cleartext character, and let r_i – random number resulted from PRNG after applying GA. Suppose the function Φ_k is the result of division α_i by r_i, denote the quotient n_i, and the remainder of this division δ_i. The function $\Phi_k(\alpha_i)$ in our system is mapping a character α_i by an ordered pair of integers n_i and δ_i. The function Φ_k, when the splitting level $k=2$, is determined by the following relation:

$$\Phi_2(\alpha_i) = \left(\left[\frac{r_i}{\alpha_i}\right]; \delta_i\right).$$

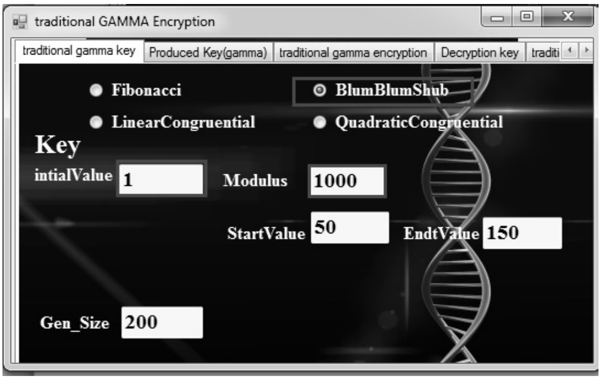
Theorem. *The mapping function* Φ_2 *at* $r_i \geqslant 256$ *reversible, and it is monomorphism.*

5. Experimental Results

5.1. A comparison between the traditional XOR encryption and the proposed one

5.1.1. The traditional method of XOR algorithm

The following example shows the restriction of the security level in the traditional XOR encryption algorithm. If the private key is selected as shown in fig. 5, the generated cryptographic keys contain only one single value $1, 1, 1, \ldots$.



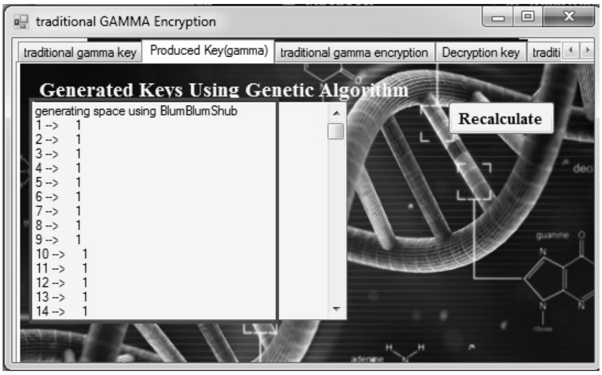
● Fibonacci
◉ BlumBlumShub
● LinearCongruential
● QuadraticCongruential

intialValue 1
Modulus 1000
StartValue 50
EndtValue 150
Gen_Size 200

Figure 5. The secret key of the traditional XOR encryption algorithm

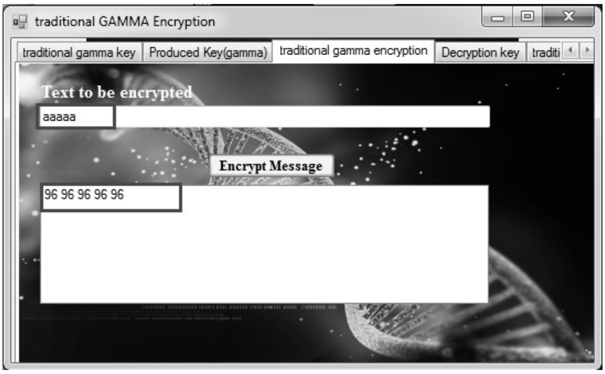
As shown in fig. 6. when encrypting the following cleartext, which consists of one character "*aaaaa*".

As shown in fig. 7, the ciphertext will have the same value for each encrypted symbol = 96 96 96 96 96, as shown in fig. 7.


traditional GAMMA Encryption
traditional gamma key
Produced Key(gamma)
traditional gamma encryption
Decryption key
traditi
Generated Keys Using Genetic Algorithm
Recalculate

```
generating space using BlumBlumShub
1 -->  1
2 -->  1
3 -->  1
4 -->  1
5 -->  1
6 -->  1
7 -->  1
8 -->  1
9 -->  1
10 -->  1
11 -->  1
12 -->  1
13 -->  1
14 -->  1
```

Figure 6. The set of cryptographic keys in the traditional XOR encryption



Text to be encrypted: aaaaa

96 96 96 96 96

Figure 7. Cleartext and ciphertext (the traditional XOR encryption algorithm)

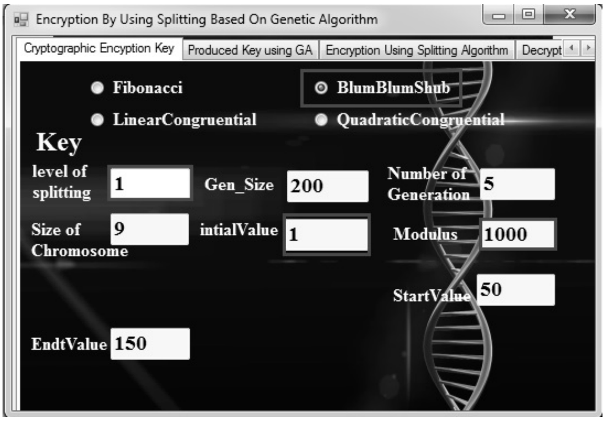
From this example, it is clear the limitation of the traditional XOR encryption algorithm, leading to relatively easily to analysis and break by attacker.

5.1.2. Modified algorithm of XOR encryption (splitting algorithm when $k = 1$).

The following example shows the improvement in the level of security, that the proposed splitting algorithm provides to the modified XOR encryption algorithm. The example is conducted by the usage of the same parameters as used in the example above. If the private key is selected as shown in fig. 8 (similar to the secret key in fig. 5), then the generated cryptographic keys contain many different values, as shown in fig. 9.

And when encrypting the cleartext, which consists of one symbol "*aaaaa*" as shown in fig. 10, the ciphertext will have several different values for each encrypted character, not one; In this case, the ciphertext will have the form "185 273 361 369 112" as shown in fig. 10. (Recall that in fig. 7, the ciphertext has the same value for all characters).

This example shows that the splitting algorithm that we proposed is highly resistant and provides a high level and degree of security as the key is variable and ciphertext vary randomly for each ciphered letters.


Encryption By Using Splitting Based On Genetic Algorithm
Cryptographic Encyption Key
Produced Key using GA
Encryption Using Splitting Algorithm
Decrypt

○ Fibonacci ● BlumBlumShub

○ LinearCongruential ○ QuadraticCongruential

Key

level of splitting: 1

Gen_Size: 200

Number of Generation: 5

Size of Chromosome: 9

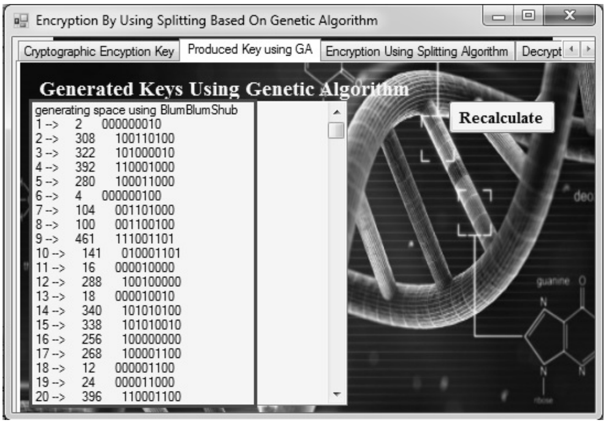
intialValue: 1

Modulus: 1000

StartValue: 50

EndtValue: 150

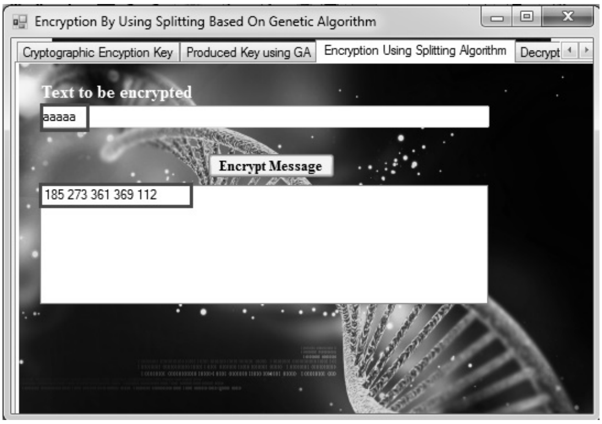
Figure 8. Secret key of the modified XOR encryption algorithm


Encryption By Using Splitting Based On Genetic Algorithm
Cryptographic Encyption Key
Produced Key using GA
Encryption Using Splitting Algorithm
Decrypt
Recalculate

Generated Keys Using Genetic Algorithm

```
generating space using BlumBlumShub
1 -->   2    000000010
2 -->   308    100110100
3 -->   322    101000010
4 -->   392    110001000
5 -->   280    100011000
6 -->   4    000000100
7 -->   104    001101000
8 -->   100    001100100
9 -->   461    111001101
10 -->   141    010001101
11 -->   16    000010000
12 -->   288    100100000
13 -->   18    000010010
14 -->   340    101010100
15 -->   338    101010010
16 -->   256    100000000
17 -->   268    100001100
18 -->   12    000001100
19 -->   24    000011000
20 -->   396    110001100
```

Figure 9. The set of cryptographic keys in the modified XOR encryption algorithm


Encryption By Using Splitting Based On Genetic Algorithm
Cryptographic Encyption Key
Produced Key using GA
Encryption Using Splitting Algorithm
Decrypt
Encrypt Message

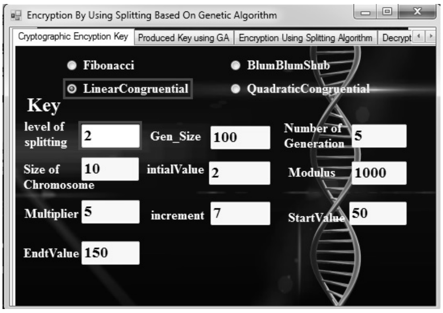
Text to be encrypted: aaaaa

185 273 361 369 112

Figure 10. Cleartext and plaintext in the modified XOR encryption algorithm

5.2. Example of applying the encryption algorithm based on the splitting procedure when splitting level k = 2

For the experiment has been selected the plaintext: $\ll$ *Encryption* $\gg$. When select a linear congruential generator based on the secret key as shown in fig. 11, we get the ciphertext $= 8\,96\,8\,24\,3\,78\,3\,18\,4\,45\,4\,75\,4\,64\,4\,96\,5\,99\,7\,7$ as shown in fig. 12.

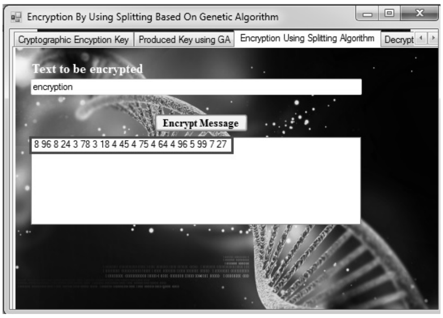

Encryption By Using Splitting Based On Genetic Algorithm
Cryptographic Encyption Key
Produced Key using GA
Encryption Using Splitting Algorithm
Decrypt

● Fibonacci ● BlumBlumShub
⊙ LinearCongruential ● QuadraticCongruential

Key

level of splitting 2
Gen_Size 100
Number of Generation 5
Size of Chromosome 10
intialValue 2
Modulus 1000
Multiplier 5
increment 7
StartValue 50
EndtValue 150

Figure 11. The secret key (splitting level = 2)


Encryption By Using Splitting Based On Genetic Algorithm
Cryptographic Encyption Key
Produced Key using GA
Encryption Using Splitting Algorithm
Decrypt
Encrypt Message

Text to be encrypted

encryption

8 96 8 24 3 78 3 18 4 45 4 75 4 64 4 96 5 99 7 27

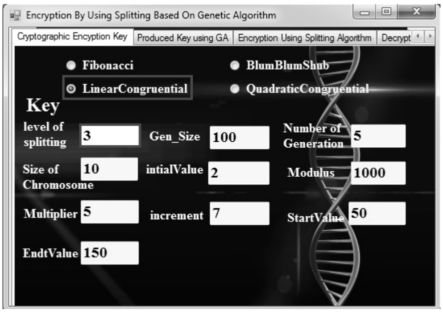
Figure 12. The cipher text in accordance with the secret key (splitting level = 2)

5.3. Example of applying the encryption algorithm based on the splitting procedure when splitting level $k = 3$

If the splitting level $k = 3$ for the same secret key as shown in fig. 13 and the same plaintext as shown in fig. 14. We'll get the cipher

$$text = 113\,0\,96\,113\,0\,24\,125\,0\,78\,120\,0\,18\,132\,1\,45\,130\,3\,75\,132\,0\,64\,129\,0\,96\,130\,4\,99\,113\,6\,27\,,$$

as shown in fig. 14.

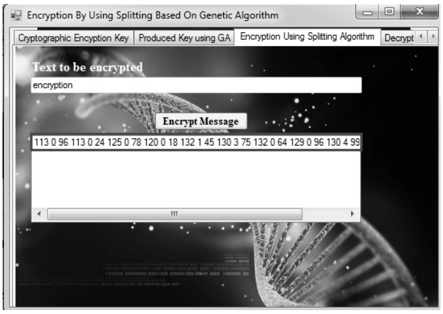

Encryption By Using Splitting Based On Genetic Algorithm
Cryptographic Encyption Key
Produced Key using GA
Encryption Using Splitting Algorithm
Decrypt

● Fibonacci ● BlumBlumShub
⊙ LinearCongruential ● QuadraticCongruential

Key

level of splitting 3
Gen_Size 100
Number of Generation 5
Size of Chromosome 10
intialValue 2
Modulus 1000
Multiplier 5
increment 7
StartValue 50
EndtValue 150

Figure 13. The secret key ($splittinglevel = 3$)


Encryption By Using Splitting Based On Genetic Algorithm
Cryptographic Encyption Key
Produced Key using GA
Encryption Using Splitting Algorithm
Decrypt
Encrypt Message

Text to be encrypted

encryption

113 0 96 113 0 24 125 0 78 120 0 18 132 1 45 130 3 75 132 0 64 129 0 96 130 4 99

Figure 14. The ciphertext in accordance with the secret key ($splittinglevel = 3$)

References

1. A. V. Sokolov, V. F. Shangin, Data protection in Distributed Enterprise Networks and Systems, DMK Press, Moscow, 2002, in Russian.

2. B. Y. Ryabko, A. N. Fionov, The Foundations of Modern Cryptography for Specialists in Information Technologies, Scientific World, Moscow, 2004, in Russian.
3. A. X. Alhussain, Cryptosystem for Providing Secured Application based on Genetic Algorithm, in: International Research Conference on Engineering, Science and Management 2014 (IRCESM 2014), International Journal of Emerging Technology and Advanced Engineering, June 2014, Dubai, United Arab Emirate, Vol. 14, Special Issue 5, 2014, pp. 8–14.
4. A. Alhussain, Symmetric Encryption Algorithm Using a Genetic Algorithm and Pseudorandom Number Generators, Natural and Technical Sciences (7(85), pages = 73–79, note = in Russian, language = english).

УДК 004.056.55

Симметричное шифрование на основе метода расщепления

В. Л. Стефанюк [*][†], А. Х. Алхуссаин[†]

[*] *Институт проблем передачи информации РАН, Москва, Россия*

[†] *Российский университет дружбы народов, Москва, Россия*

Предлагаемое в статье расщепление касается защищенных способов передачи информации о каждом отдельном символе при их потоковой передаче. Расщепление является обобщением процедуры защищенной передачи с одиночным ключом.

Интеллектуальность расщепления состоит в обратимом кодировании (reversible coding) отдельных символов вместо использования теоремы отсчетов Котельникова-Шеннона для передачи по каналу связи последовательности блоков символов, с целью повышения степени защиты. Описан действующий вариант системы, предназначенный для передачи текстовых сообщений. В этой статье показано, как использовать набор криптографических ключей, которые генерируются с применением генетического алгоритма, и как выбрать генератор псевдослучайных чисел, чтобы решить некоторые современные криптографические задачи.

Ключевые слова: генетический алгоритм, генератор псевдослучайных чисел, шифрование, дешифрование, криптография, мономорфизм, алгоритм расщепления

Литература

1. *Sokolov A. V., Shangin V. F.* Data protection in Distributed Enterprise Networks and Systems. — Moscow: DMK Press, 2002. — In Russian.
2. *Ryabko B. Y., Fionov A. N.* The Foundations of Modern Cryptography for Specialists in Information Technologies. — Moscow: Scientific World, 2004. — In Russian.
3. *Alhussain A. X.* Cryptosystem for Providing Secured Application based on Genetic Algorithm // International Research Conference on Engineering, Science and Management 2014 (IRCESM 2014), International Journal of Emerging Technology and Advanced Engineering, June 2014, Dubai, United Arab Emirate. — Vol. 14, Special Issue 5. — 2014. — Pp. 8–14.
4. *Alhussain A.* Symmetric Encryption Algorithm Using a Genetic Algorithm and Pseudorandom Number Generators // Natural and Technical Sciences. — 2015. — No 7(85), pages = 73–79, note = in Russian, language = english.

© Stefanyuk V. L., Alhussain A. H., 2016