ISSN 1512-1232

New block encryption algorithm

Zurab Kochladze

Ivane Javakhishvili Tbilisi State UniversityFaculty of Exact and Natural Sciences, Department of Computer Science0186, University street 13. Tbilisi, Georgia,
zurab.kochladze@tsu.ge

Abstract
This paper describes a new block encryption algorithm that uses the Hill's modified algorithm for faster efficiency process. This allows us to increase the encryption and decryption speeds so as not to reduce the algorithm's resistance to cryptanalytic attacks.

Keywords: *Symmetrical algorithms, block cipher, Hill's modified algorithm*.

I. Introduction

Modern block algorithms are very often very substantially different from each other [1, 2] in both, architecture and the number of operations and rounds, but the outcome of their work is always the same. The starting line is a binary string with the length of n, whose structure is defined by the open text, by the key of length k and the use of certain operations, after the multiple iterations goes back to the n length pseudo-random bit string. In fact, any block algorithm mathematically can be imagined as function of two variables

$$E:\{0,1\}^n \times \{0,1\}^k \to \{0,1\}^n,$$

where $(0,1)^l$ notes bit string of l length, and k and n values depend on the specific of encryption algorithm. In practice, this means that for each fixed $K \in \{0,1\}^k$ encryption function is a replacement on $\{0,1\}^n$ bit string [3, 4]. Obviously, the received function can't be absolutely random, since the transfer is done with the determinant algorithm. This means that any such algorithm can theoretically be broken, and it can be only computationally protected against the cryptanalytic attacks. In order to prevent an_opponent with limited computational resources from breaking the algorithm, it is essential that the binary string that is encrypted by encryption algorithm, will be near with a random binary string.

As it is well known, C. Shannon in his fundamental work [5] showed that to achieve this goalit is necessary, that maximal number of open-text symbols to take part in getting one symbol of cipher-text. To achieve this goal, modern bloc ciphers use several Iterations, i.e. the same block is encoded several times using different keys. Obviously, repeatting the same procedure increases the encryption time. Thus, it is better that the operations used in the rounds are more effective in this regard.

II. Description of algorithm

In 1930, the American mathematician L.S. Hill developed the previously existing bigram and trigram ciphers and introduced a n-gram encryption using a linear algebra [6]. The essence of the algorithm is that as it is obtained in the classical cryptography, the letters of the encrypted text will

ISSN 1512-1232

be transferred to the numbers. Then these numbers are divided into vectors of length and are multiplied to a $n \times n$ square matrix by module n, where n is the number of characters in the language on which the open text is drawn. The matrix that represents the key of this algorithm must have a reverse matrix. It is not easy to use only crypto-text to attack the algorithm, but it is easy to attack using open-text, because the conversion is a straight line, and if the size of the matrix is $n \times n$, then only the linear n^2 equation system is needed to accurately calculate the key. Because of these reasons, the long-term algorithm was no longer used in computer cryptography, although the multiplication operation on the matrix has a very high efficiency of diffusion. In recent years the works [7,8,9,10] have been published, the authors of which are still trying to use different options of the Hill's Algorithm due to the quality.

In the articles [11,12], the author describes Hill's modified algorithm that can be used in cipher in which the encrypted block can be viewed as a matrix of the condition (for example AES standard [13]). This article discusses a new block algorithm that uses the modified version of Hill's algorithm (fig. 1).

Description of the algorithm: the size of the block is 128 bits. Two keys are used for encryption, each of them is 128 bits long. The open text will be viewed as ASP-II codes in binary string and will be divided into 128 bit length blocks. Before the open text will enter in first round, it gathers with the 128-bit first key with the xor operation. Each round consists with three operations: multiplication on the self-reversible matrix, shifting the bytes in matrix and gathering with the round key. The result of first operation will be divided by 16 bits (16 bytes) and will be written as a square matrix (4×4). Recording from left to right and down from the top. Bytes will be transferred in decimal systems. Received matrix is multiplied by the self-reversible matrix by mod256. In received matrix, the bytes are shifted to left by strings by one byte. The bits string will be transformed into a matrix and will enter the second round matrix will be transfer on bit string and we gather it with the second key by xor operation. The gathered bits string will be transformed into a matrix and will enter the second round. After four rounds, we get an encrypted text.

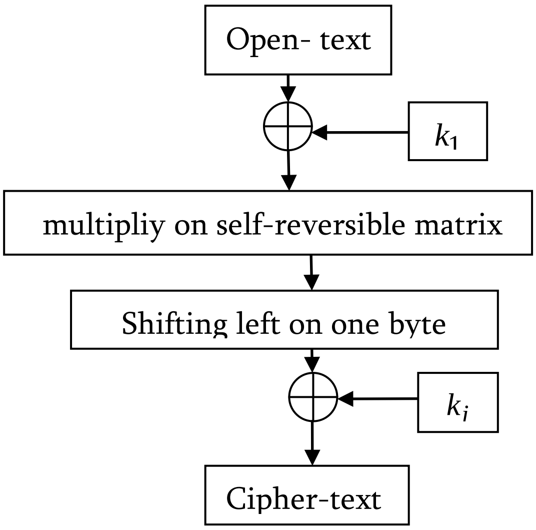

Fig.1.

To get round keys we take a random matrix and multiply it on the previous round key. So $k_i = k_{i-1} \times A (\mathrm{mod}\, 256),$ where A is a random matrix.

Consider an example. Let's assume you have the text of the encryption: "domain parameters". Self-reversible matrix

ISSN 1512-1232

$$\begin{pmatrix} 2 & -1 & -2 & 2 \\ -1 & -2 & -2 & -2 \\ 1 & 1 & 1 & 2 \\ -1 & 1 & 2 & -1 \end{pmatrix}.$$

Two key:

K =
00101111 10010101 01011011 10000010 00010010 11010110 10101011 11010111
01101101 11000010 11101100 10011001 01101010 1 0101010 10000100 10101001.
K_1 =
10101101 00110100 10010010 01000100 10000001 01001011 01011100 11101010
10101010 01011000 01001000 11010100 11101000 10101010 10101010 10000111

A random matrix for getting the round keys is

$$\begin{pmatrix} 102 & 98 & 212 & 179 \\ 85 & 211 & 146 & 221 \\ 155 & 76 & 231 & 166 \\ 39 & 128 & 150 & 29 \end{pmatrix}.$$

Convert the key K_1 to the matrix we get:

$$\begin{pmatrix} 173 & 52 & 146 & 64 \\ 129 & 75 & 92 & 234 \\ 170 & 88 & 72 & 212 \\ 232 & 170 & 170 & 135 \end{pmatrix}$$

After the computations, we obtain, that

K_2= 10111100 01100110 00101010 11000111 10100111 10000011 10000010 00010010
11011000 11111100 00101000 10001010 01100001 11100110 10010100 10000001,
K_3= 10000101 00000010 01011100 01001001 01111101 01111111 11011100 00110010
11111010 01000100 11001010 00100100 01000111 00100100 10100010 11110110
K_4= 01111011 01100000 00010010 10100110 11001011 11010111 01011110 01011100
01001000 11001110 00100000 11011100 11001110 11110010 10100110 10100011.

Now we convert open text in bits string and gather it with key K. Then we'll turn the bits string into matrix and get it:

$$\begin{pmatrix} 75 & 250 & 54 & 227 \\ 123 & 184 & 219 & 182 \\ 31 & 227 & 229 & 252 \\ 30 & 207 & 230 & 218 \end{pmatrix}$$

ISSN 1512-1232

Then we multiply the received matrix by the self-reversible matrix and we get:

$$\begin{pmatrix} 239 & 201 & 227 & 23 \\ 99 & 66 & 225 & 134 \\ 68 & 151 & 77 & 214 \\ 121 & 4 & 192 & 144 \end{pmatrix}$$

Shifting

$$\begin{pmatrix} 201 & 226 & 23 & 239 \\ 66 & 225 & 134 & 99 \\ 151 & 77 & 214 & 68 \\ 4 & 192 & 144 & 121 \end{pmatrix}$$

Then we convert the received matrix to the bits string and gather the first keys. This will be the first round output:

11100100 11010110 10000101 10101011 11000011 10101010 11111010 10001001
00111101 00010101 10011110 10010000 11101100 01101010 00111010 11111110

III. Conclusion.

The described algorithm satisfies all the features, nec

essary for modern symmetric algorithms and is very fast that will allow us to use this algorithm to encode large texts.

References:

[1] B. Schneier. Applied cryptography, John Wiley & Sons, Inc. 1996. (?)

[2] M. Mogollon. Cryptography and Security Services. Cybertech Publishing 2007

[3] R. Oppilger. Contemporary Cryptography Artech House Boston/ London 2005.

[4] M. Bellare, P. Rogaway. Introduction to Modern Cryptography. -UCSD CSE 207, 2005.

[5] C. Shannon. Communication theory of secrecy systems. Bell System tech. J., 28, #4 (1949), 656-715.

[6] Lester S. Hill. Cryptography in an algebraic alphabet. //The American Mathematical Monthly, vol.56, #6, 1929, pp. 306-312.

[7] Bibhudendra Acharya, Sarojkumar Panigrahy, Saratkumar Patra, Canapsti Panda. Image Encryption Using Advanced Hill Cipher Algorithm. // International Journal of Recent Trends in Engineering, May 2009, vol.1, No.1, pp.663-667.

[8] Bibhudendra Acharya, SarojkumarPanigrahy, SaratkumarPatra, and Canapsti Panda. Image Encryption Using Advanced Hill Cipher Algorithm. International Journal of Recent Trends in Engineering. Vol.1, No.1, May 2009. pp.663-667.

[9] M. Farmanbar, A.G. Chefranov. Investigation of Hill Cipher Modifications Based on Permutation and Iteration. (IJCSIS) International Journal of Computer Science and Information Security. Vol.10, No9, September, 2012. pp.1-7.

[10] V.U.K.Sastry, A.Varanasi and S.U. Kumar. A Modern Advanced Hill Cipher Involving a Permuted Key and Modular Arithmetic Addition Operation

[11] Z. Kochladze. Modified Version of the Hill’s Algorithm. GESJ: Computer Science and Telecommunication No3 (43) 2014.

ISSN 1512-1232
[12] L. Djulakidze, Z. Kochladze, T. Kaishauri. New symmetrical tweakable block cipher. Georgian engineering innovations, No.1 (vol.73), 2015, გვ.50. In Georgian.
[13] J. Daemen, V. Rijmen The Decign of Rijndael: AES – The Advanced Encryption Standard. Springer, 2002.

Article received: 2018-01-16