ISSN 1512-1232

Modified Version of the Hill's Algorithm

Zurab Kochladze

Department of Computer Sciences, Iv.Javakhishvili Tbilisi State University, 13,
University str., 0186 Tbilisi, Georgia

Abstract

The paper discusses the modified variant of Hill's algorithm that can be used in the modern block ciphers.

Keywords*: Symmetrical block cipher, Hill's algorithm, The principle of confusion and diffusion.*

I. Introduction

It's well recognized that due to the very low speed of the public key cipher, to protect the confidentiality of information the symmetric block cipher is commonly used. The block algorithms are usually significantly different from each other both in terms of the architecture, as well as the operations and the number of rounds; However, the result of their work is always the same, n length of the bit string, whose structure has been determined by open text, using the k length key, which is also k length of the bit string, and some operations and using multiple iterate moves back to the n length of the pseudorandom bit string. In fact, mathematically, we can consider any block algorithm as a function of two variables

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

where $\{0,1\}^l$ represents the bit string of l length, while the values of k and n depends on the concrete encryption algorithm.

Practically for each fixed $K \in \{0,1\}^k$ encrypting function represents a permutation on the set $\{0,1\}^n$. As it is well known, C. Shannon in his fundamental work [4] showed that there is only one theoretically unbreakable symmetric cipher (One-Time Pad) of this type, whose successful operation requires implementation of the following conditions: The length of the key must be equal the duration of the open text, the key must be a totally random sequence, and the key must be used only once (It's why this cipher called one-time pad). It is a clear that the use of this type of the cipher in everyday practice is very inconvenient, therefore in practice is widely used the symmetric algorithms that are only computationally secure against attacks of the adversary. This means that if the adversary has unlimited computationally abilities, he can always break a cipher, but in practice there is no unlimited computationally capable opponent. Therefore, it is important to find quantitative relations between the adversary's capabilities and the resistance of the cipher that gives us possibility to assess quantitatively the security of symmetrical block ciphers against attacks. In present paper two cases of the attack will be reviewed.

If cryptanalytics's goal is to calculate the key, then the security analysis of the block ciphers can be formulated in following form [5]: Given the encryption function $E_k(M)=C$, where $K \in \{0,1\}^k$ is an unknown key. The cryptanalytic knows entry and exit $(M_1,C_1),...(M_q,C_q)$ values for any q number of pairs, and he or she is trying to calculate the key. In this case, the block cipher will be secure, if the best possible attack carried by the opponent will require a large number of q couples and/or the calculation will take a great t time, that exceeds the abilities of any cryptanalytic. This is a security against key recovery and measured quantitatively by the parameters q and t.

Nevertheless, the fact that the block cipher will be secure against the attacks of key recovery, does not necessarily means that it will be secure in general. As C. Shannon showed in the same

ISSN 1512-1232

report, the algorithm might allow leakage of any sort of information about open text. If the encoding algorithm allows the leaking of this type of data, then cryptanalytic have a fortune to collect a certain quantity of data and break the algorithm, or restore the open text.

Therefore to ensure the security of the cipher for a long time, we should prove that by the computational tools that are employed by the opponent, it is impossible to receive any kind of data about open text. This signifies that the encryption algorithm should well cover the open text structure in cipher-text. To hide the open text structure in cipher-text the most effective means are using two transformations - confusion and diffusion. The confusion is the transformation, that aims to cover the connections between the key and the cipher-text, while the diffusion aims to ensure that each character of the cipher-text is dependent to the all characters of the open text. It gives us mean to hide the open text structure in cipher-text.

The usage of the complex mathematician transformation in the symmetrical algorithms is not recommended, as it lowers the speed of the algorithm. Therefore, to achieve the same goals in modern symmetrical cryptology generally use substitution and displacement operations. In order to achieve the desired level of diffusion, on the block that should be encrypted undergoes the same operation several times with different keys. The cycle of the operations called round. Obviously, the greater is the number of rounds; the speed of the algorithm is lower.

The paper reviews the modification of the well-known Hill's algorithm [7] that allows a very fast implementation of the diffusion transformation, which we believe will reduce the number of rounds.

II. Hill's algorithm

In 1929, American mathematician Lester S. Hill invented polygraphic substitution chipper based on linear Algebra. In this particular chipper, the any output symbol in the algorithm depends on all n input symbols. Hill corresponds symbols of the open text number from zero to 26 as it has been used in most of the chippers in classic cryptology. The open text was transformed in numbers and divided into n length blocks. To encrypt a message each block of n letters (considered as an n-component vector) is multiplied by square $n \times n$ matrix, again modulus 26. As a result, received n length vector, that represents the cipher-text and its each symbol is dependent on input vector n symbol.

The above mentioned is the most important and critical difference of Hill's algorithm from previously existing encryption methods. In order to decrypt, the encryption matrix should have inverse matrix module26. For this purpose, it is enough that the matrix determinant differs from zero and be coprime at the root of the module.

For example, if we want that one output symbol of the cipher-text is dependent on three symbols of the open text, we should take the matrix 3×3, such

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} mod\ 26$$

That $A \cdot A^{-1} = E$, where E is the unit matrix and multiply it on open text trigram (the numbers after the transfer)

$$M \times A = C(mod\ 26).$$

the decryption formula will looks like :

$$. \ C \times A^{-1} = M(mod\ 26)$$

Apparently, the larger encryption matrix size is, the more letters of the open text will take a part in calculation of one output symbol of cipher text. And more open text structure will be well hidden in the cipher-text. However, Hill's algorithm usage by hand encryption is very difficult, and therefore encryption matrix size is low, that makes difficult to achieve a set goal.

ISSN 1512-1232

During the first phase of development of computer cryptography the use of the Hill's algorithm was rejected. The explanation was that the vector matrix multiplication is a linear operation and if the algorithm used $n \times n$ the matrix, to break it the only needed to solve n^2 linear equation. However, in recent years the number of studies appeared [8,9] that use the various modification of Hill's algorithm together with a nonlinear operation. It makes impossible to easily break the algorithm and it retains all the good characters of Hill algorithm.

III. Modified version of Hill's algorithm

The presented modified Hill's algorithm can be used in ciphers where encryption block is represented as state matrix (e.g. AES standard). Let's consider the cipher, where the block size equals to 128 bits. This block in Algorithm can be represented as a 4×4 matrix, so called state matrix.

$$M = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$$

Here, each a_{ij} representing a binary bytes. Encrypted binary string written horizontally from left to right in the matrix if transformed o each a_{ij} in decimal system, for any element of a_{ij} fulfilled promise $0 \le a_{ij} \le 255$.

In the modified algorithm the multiplication of the state matrix on 4×4 dimension A matrix , again module 256 takes place.

$$M \times A (mod\ 256).$$

It is obvious that the multiplication of the matrix on the matrix is not a simple operation, such as displacement or replacement, therefore it may substantially affect the speed of the algorithm. In order to keep the algorithm speed within acceptable limits, the matrix elements should be the smallest numbers as possible. In this case it's possible that invertible matrix elements represent a large numbers that will increase the decryption time. Given these reasons, we have selected self-invertible matrix

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

The elements of the matrix are only ± 1 and ± 2 numbers. (It gets easier to multiply matrix on matrix operation). The fact that multiplication of such matrix results in state matrix negative numbers does not represent a problem, as simply, through the addition of a module it can be transformed in positive numbers again.

Let's consider the following example. The given open text is "domain parameters". Through ASCII codes correspond the letters the numbers in decimal system and transform it in the bit string. We will get matrix 4×4

$$M = \begin{pmatrix} 100 & 111 & 109 & 97 \\ 105 & 110 & 112 & 97 \\ 114 & 97 & 109 & 101 \\ 116 & 101 & 114 & 115 \end{pmatrix}$$

and bit string:

01100100 01101111 01101101 01100001 01101001 01101110 01110000 01100001 01110010 01100001 01101101 01100101 01110100 01100101 01110010 01110011.

Multiply resulted matrix M on A matrix again module 256. We will get there

ISSN 1512-1232

$$\begin{pmatrix} 100 & 111 & 109 & 97 \\ 105 & 110 & 112 & 97 \\ 114 & 97 & 109 & 101 \\ 116 & 101 & 114 & 115 \end{pmatrix} \times \begin{pmatrix} 2 & -1 & -2 & 2 \\ -1 & -2 & -2 & -2 \\ 1 & 1 & 1 & 2 \\ -1 & 1 & 2 & -1 \end{pmatrix} - mod\ 256 = \begin{pmatrix} 101 & 140 & 137 & 99 \\ 115 & 140 & 132 & 117 \\ 139 & 158 & 44 & 151 \\ 130 & 157 & 166 & 143 \end{pmatrix} mod\ 256,$$

It will give the bit string:

01100101 10001100 10001001 01100011 01110011 10001100 10000100 01110101 10001011 10011110 00101100 10010111 10000010 10011101 10100110 10001111.

IV. The conclusion.

Let's compare the initial and resulted bit strings with each other.
The starting line was:

01100100 01101111 01101101 01100001 01101001 01101110 01110000 01100001 01110010 01100001 01101101 01100101 01110100 01100101 01110010 01110011

The resulted string is:

01100101 10001100 10001001 01100011 01110011 10001100 10000100 01110101 10001011 10011110 00101100 10010111 10000010 10011101 10100110 10001111.

From 128 bits the difference is among 67 bits that indicates that using of this type of the algorithm is effective mean to achieve the necessary level of diffusion rapidly. The output string bit will represents pseudorandom bit string. In addition, it needs to recognize the additional operation, key and iteration number that requires futher work. After completion of the research the output pseudorandom bit string will be non-distinct from random bit string.

References:

1. B. Schneier Applied cryptography John Wiley & Sons, Inc. 1996.
2. M. Mogollon Cryptography and Security Services. Cybertech Publishing 2007
3. Oppilger R. Contemporary Cryptography Atech House Boston|london 2005.
4. C. Shannon Communication theory of secrecy systems. Bell System tech. J.,28, №4 (1949), 656-715.
5. M. Bellare, P. Rogaway The security of Triple Encryption and Framework for Code-Based Game-playng Proofs. Eurocrypt 2006, LNCS vol. 4004, Springer, 2006, pp. 409-426.
6. Lester S. Hill Cryptography in an algebraic Alphabet. The American Mathematical Monthly Vol.56 №6 (1929) pp. 306-312.
7. 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.
8. M. Farmanbar, A.G. Chefranov Investigation of Hill Cipher Modifications Based on Permutation and Iteration. (IJCSIS) International Journal of Computer Ccience and Information Security. Vol.10, No9, September 2012. pp.1-7.

Article received: 2014-09-08