DOI: 10.24996.ijs.2017.58.2C.18

ISSN: 0067-2904

Text Compression & Encryption Method Based on RNA and MTF

Ekhlas Khalaf Gbashi
Department of Computer Science, University of Technology, Baghdad, Iraq.

Abstract

The sending of information at the present time requires the speed and providing protection for it. So compression of the data is used in order to provide speed and encryption is used in order to provide protection. In this paper a proposed method is presented in order to provide compression and security for the secret information before sending it. The proposed method based on especial keys with MTF transform method to provide compression and based on RNA coding with MTF encoding method to provide security. The proposed method based on multi secret keys. Every key is designed in an especial way. The main reason in designing these keys in special way is to protect these keys from the predication of the unauthorized users.

Keywords: Data Compression, Encryption, RNA, MTFE, MTFD, private keys.

طريقة ضغط وتشفير نص بالاعتماد على ال RNA وال MTF

اخلاص خلف كباشي

قسم علوم الحاسوب ، الجامعة التكنولوجية ، بغداد ، العراق.

الخلاصة

ان ارسال المعلومات في الوقت الحاضر يتطلب سرعة وتوفير حماية لها .لذلك يتم استخدام ضغط البيانات لتوفير السرعة والتشفير لتوفير حماية لها .في هذا البحث الطريقة المقترحة تم تقديمها لغرض توفير ضغط وامنية للمعلومات السرية قبل ارسالها .الطريقة المقترحة تعتمد على مفاتيح خاصة مع طريقة ال MTF لتوفير ضغط وتعتمد على ترميز ال RNA مع طريقة ال MTF لتوفير الامنية .ان الطريقة المقترحة تعتمد على اكثر من مفتاح وكل مفتاح مصمم بطريقة معينة. الغرض الاساسي من تصميم هذه المفاتيح بطريقة معينة هي المحافظة عليها من التنبؤ بها من قبل الاشخاص الغير مخولين.

1. Introduction

Data compression is primarily a branch of information theory [1]. The definition of Data Compression may be phrased as the encoding the data with the use of a small number of bits instead of the original representation. There are 2 kinds of compression, known as lossless and lossy. The latter type is an approach of data encoding, where the compression is performed via the discarding/eliminating a portion of the data. This is typically utilized in multimedia data, specifically in applications such as streaming media and internet telephony. In those application a certain degree of information loss might be tolerable. Dropping the non-necessary detail from the data may be beneficial in saving storage space. The lossless type of compression may be identified as the reduction of bits via the identification and elimination of the statistical redundancy. This type of compression is

Email:110026@uotechnology.edu.iq

reversible of lossy compression, in a way that the exaction of the genuine data that is to be re-constructed from the compressed data. Lossless compression may be utilized for images, audios and so on, but more commonly it's utilized for text data such as executable programs, text documents and source codes. There are several various methods for measuring the efficiency of a compression algorithm as in equation (1) and equation (2). The basic concern of performance measures is the efficiency concerning space and time. Following are a set of factors utilized for the evaluation of the efficiency of the lossless approaches [2].

$$\text{Compression Ratio} = \frac{\text{size after compression}}{\text{size before compression}} \quad \ldots\ldots (1)$$

$$\text{Compression Factor} = \frac{\text{size before compression}}{\text{size after compression}} \quad \ldots\ldots (2)$$

Security is the one of most important concerns in various types of network communications and individual countries as well. Cryptographic approaches become much more valuable for data transmitting via insecure channels [3].

Encryption is a cryptographic approach which makes it hard to interpret a document for anyone without having the knowledge of the decryption key [4].Encryption can also be defined as the operation of producing the secret text from the input text by utilizing a private key and encryption algorithm. Plain text is referred to the input text and the cipher text refers to secret text that is generated. Basically there are two categories of the encryption algorithms that are symmetrical key encryption algorithm and asymmetrical key encryption algorithm. In Symmetrical key encryption algorithm a single key is utilized by each of the sender and the receiver but in asymmetrical key algorithm sender and receiver both utilize different keys [5].

2. RNA

DNA sequences are used to encrypt information in the encryption of the communication methods, mainly the ones that need a robust data encryption scheme to challenge unauthorized access. DNA sequences, which consist of the following nitrogenous bases: (A, T,C, G), complementary to RNA sequences, which consist of the following: (U, A, G, C). The non-coding segments, called introns in DNA sequences, are removed by splicing and the remaining segments that encode information for protein synthesis, called exons, are assembled in mRNA [6]. Ribonucleic acid (RNA) is a copy of the DNA to get out to the cytoplasm to decide what the cell should do for surviving [7]. In fact, RNA is like DNA in having only four bases, while proteins could include up to 20 amino acids. Permutation of the 4 bases produces 4^3 or 64 triplets [8].

3. Move-To-Front Transform

MTFT transform algorithm is almost forgotten that may reduce the universe of information. The MTFT procedure can be described as follows [9]:

1. Forming a list L. The list L contains each unique happening character from the data stream that is entered I (i.e. L includes the I alphabet).

2. for each character $s \in I$:

(2.1) discover s location from the (list L)& output the Index of it i ($s = L[i]$).

(2.2) reduce location of each characters that starts from the (Index (0))to the(Index ($i-1$)) within L ($L[J + 1] = L[J], J = i - 1, i - 2, \ldots, 0$).

(2.3) Shift s to the front of the list L ($L[0] = s$).

It is clear, that the resulted output stream O includes the list L indexes, but the indexes number in the input stream and in output stream stay identical ($|I| = |O|$). The input stream I is effortless reconstructed by swapping every index ($i \in O$) with the character ($s = L[i]$), and shifting s to the front of the list L. This procedure considerable as the inverse move- to -front transforms (IMTFT).The configuration of L is identical in both cases: the coding phase and the decoding phase. Basic cause for implementing MTFT in the compression algorithms is that to provide the ability to decrease the information entropy. MTFT phase could be implemented in ($n \in [1, \infty]$) reiterations, where ($i + 1$) th reiteration utilizes i th reiteration as the input. After n IMTFT phase reiterations, the original stream is then retrieved. Shannon entropy (see equation (3)) is a perfect measure to find the most appropriate MTFT reiterations number for the considered input stream I:

$$H(I) = -\sum_{i=1}^{|I|} p(s)\, log_2(p(s)), \quad \text{..... (3)}$$

Where: p(s) is the possibility of happening for character s ∈ I Table-1 shows an example of the MTR transform [9].

Table 1-MTFT (left) and IMTFT (right) Example on VCC.

Step	VCC input	MTFT output	*L*
1	**2**2213131313	0	213
3	2**2**213131313	00	213
3	22**2**13131313	000	213
4	222**1**3131313	0001	123
5	2221**3**131313	00012	312
6	22213**1**31313	000121	132
7	222131**3**1313	0001211	312
8	22213131**3**13	00012111	132
9	222131313**1**3	000121111	312
12	2221313131**3**	0001211111	132
11	22213131313	000121111111	312

IMTFT input	VCC output	*L*
00012111111	2	213
0**0**012111111	22	213
00**0**12111111	222	213
000**1**2111111	2221	123
0001**2**111111	22213	312
00012**1**11111	222131	132
000121**1**1111	2221313	312
0001211**1**111	22213131	132
00012111**1**11	222131313	312
000121111**1**1	2221313131	132
0001211111**1**	22213131313	312

4. The proposed method

The proposed method provides compression and security to the secret text based on MTFE method and multi private keys, RNA coding. The compression ratio achieved to 0.75 based on MTF method since each 8 bits will be swapped with 6 bits. The high security that the proposed method achieved based on using the features of common encryption methods like permutation feature according to specific key like DES method and others features. The high security of the proposed method also comes from using multi private keys, since five keys are used and every key has multi potentials which prevent the attacker from knowing the original private keys that are used for the encryption of secret text. The first key is a table consists of 64 elements designed in predefined way between the sender and the receiver where each element has 64 potentials. The second key it's length is 6 numbers these numbers are predefined between the sender and the receiver. The third key is table consists of two columns, the first column will contains binary codes (6 bits) and the second column will contains RNA codon. The fourth key it's length is 3 numbers to make the permutation to RNA codon. The last is a dictionary used to convert from the decimal numbers to binary numbers.

The following two figures and algorithms show the proposed method in both sides: the sender side and the receiver side.

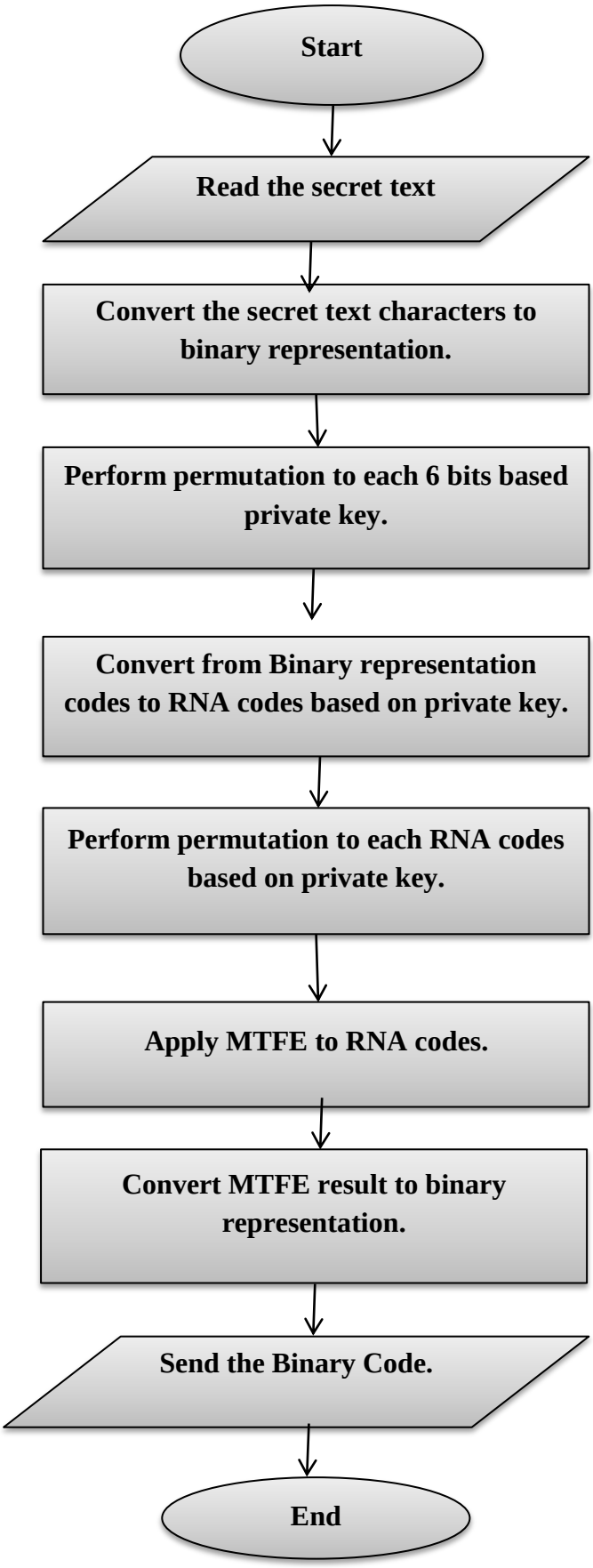

Figure 1- the proposed method at the sender side.

The sender side algorithm
Input:- The secret text , the five private keys.
Output:- The ciphertext
Begin **Step1**:- Read the secret text. **Step2**:- Convert the secret text characters to binary representation using special dictionary (private key1 which is dictionary contains 64 elements, every element is 6 bits). **Step 3**:- Perform permutation to each 6 bits according to special private key 2 its length is 6 bits numbers. **Step 4**:- perform convert from the binary codes to RNA codes using private key3 which is special dictionary. **Step 5**:- Perform permutation to each RNA codes (3 codes) according to special keys (private key 4) its length is3 numbers. **Step 6**:-Apply MTFE to RNA codes. **Step 7**:-Convert MTFE decimal numbers to binary codes use private key5. **Step 8**:- Send the binary code. **End**

Table 2- Private Key1

Seq.	Character	Binary Representation	Seq.	Character	Binary Representation
0	A	000000	32	g	100000
1	B	000001	33	h	100001
2	C	000010	34	i	100010
3	D	000011	35	j	100011
4	E	000100	36	k	100100
5	F	000101	37	l	100101
6	G	000110	38	m	100110
7	H	000111	39	n	100111
8	I	001000	40	o	101000
9	J	001001	41	p	101001
10	K	001010	42	q	101010
11	L	001011	43	r	101011
12	M	001100	44	s	101100
13	N	001101	45	t	101101
14	O	001110	46	u	101110
15	P	001111	47	v	101111
16	Q	010000	48	w	110000
17	R	010001	49	x	110001
18	S	010010	50	y	110010
19	T	010011	51	z	110011
20	U	010100	52	0	110100
21	V	010101	53	1	110101
22	W	010110	54	2	110110
23	X	010111	55	3	110111
24	Y	011000	56	4	111000
25	Z	011001	57	5	111001
26	a	011010	58	6	111010
27	b	011011	59	7	111011
28	c	011100	60	8	111100
29	d	011101	61	9	111101
30	e	011110	62	.	111110
31	f	011111	63	,	111111

- **Private Key 2**
  324516

Table 3- Private Key3

Seq.	Binary Representation	RNA Codes	Seq.	Binary Representation	RNA Codes
1	111111	'AAA'	33	011111	'GAA'
2	111110	'AAC'	34	011110	'GAC'
3	111101	'AAG'	35	011101	'GAG'
4	111100	'AAU'	36	011100	'GAU'
5	111011	'ACA'	37	011011	'GCA'
6	111010	'ACC'	38	011010	'GCC'
7	111001	'ACG'	39	011001	'GCG'
8	111000	'ACU'	40	011000	'GCU'
9	110111	'AGA'	41	010111	'GGA'
10	110110	'AGC'	42	010110	'GGC'
11	110101	'AGG'	43	010101	'GGG'
12	110100	'AGU'	44	010100	'GGU'
13	110011	'AUA'	45	010011	'GUA'
14	110010	'AUC'	46	010010	'GUC'
15	110001	'AUG'	47	010001	'GUG
16	110000	'AUU'	48	010000	'GUU'
17	101111	'CAA'	49	001111	'UAA'
18	101110	'CAC'	50	001110	'UAC'
19	101101	'CAG'	51	001101	'UAG'
20	101100	'CAU'	52	001100	'UAU'
21	101011	'CCA'	53	001011	'UCA'
22	101010	'CCC'	54	001010	'UCC'
23	101001	'CCG'	55	001001	'UCG'
24	101000	'CCU'	56	001000	'UCU'
25	100111	'CGA'	57	000111	'UGA'
26	100110	'CGC'	58	000110	'UGC'
27	100101	'CGG'	59	000101	'UGG'
28	100100	'CGU'	60	000100	'UGU'
29	100011	'CUA'	61	000011	'UUA'
30	100010	'CUC'	62	000010	'UUC'
31	100001	'CUG'	63	000001	'UUG'
32	100000	'CUU'	64	000000	'UUU'

- **Private Key 4**
  231

Table 4- Private Key1

Decimal Number	Binary representation
0	**00**
1	**01**
2	**10**
3	**11**

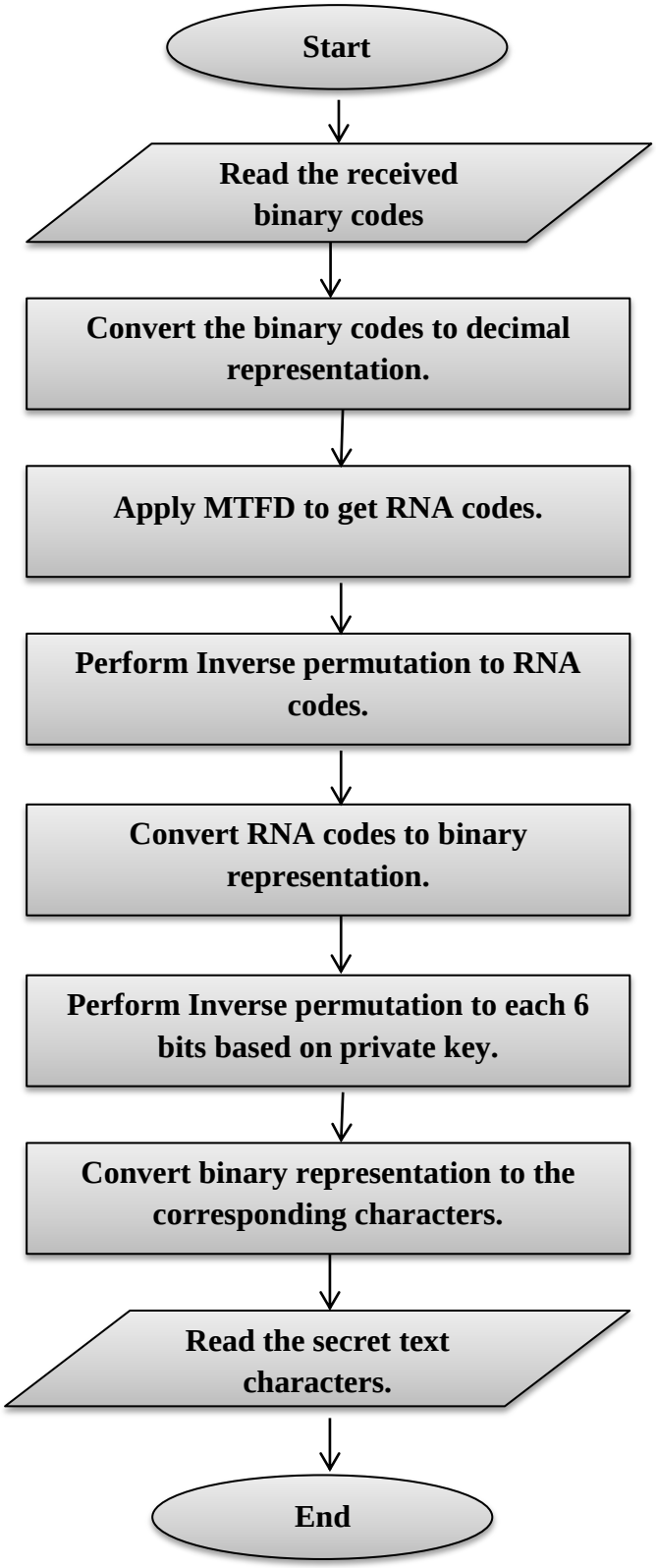

Figure 2- The proposed method at the receiver side

• **The Receiver side algorithm**
Input:- The ciphertext , the private keys .
Output:- The secret text
Begin **Step1**:- Receive the binary code. **Step2**:- Convert the binary codes to decimal values use private key5. **Step3**:- Apply MTFD to get RNA codes. **Step4**:-Perform inverse permutation to RNA codes (for each 3 codes) use private key4. **Step5**:-Convert from RNA codes to binary codes using special dictionary (private key3). **Step6**:-Perform inverse permutation to each (6 bits) use private key2. **Step7**:-Convert binary representation to corresponding characters use private key1. **Step8**:-Read the secret text characters. **Step 10**:- End.

5. Case Study of the Proposed Method

In this section two examples will be explained in details to show the steps of the proposed method. Consider the secret text that we want to encrypt it is "text". According to the steps of the encryption it will processed as follow in the Figure -3.

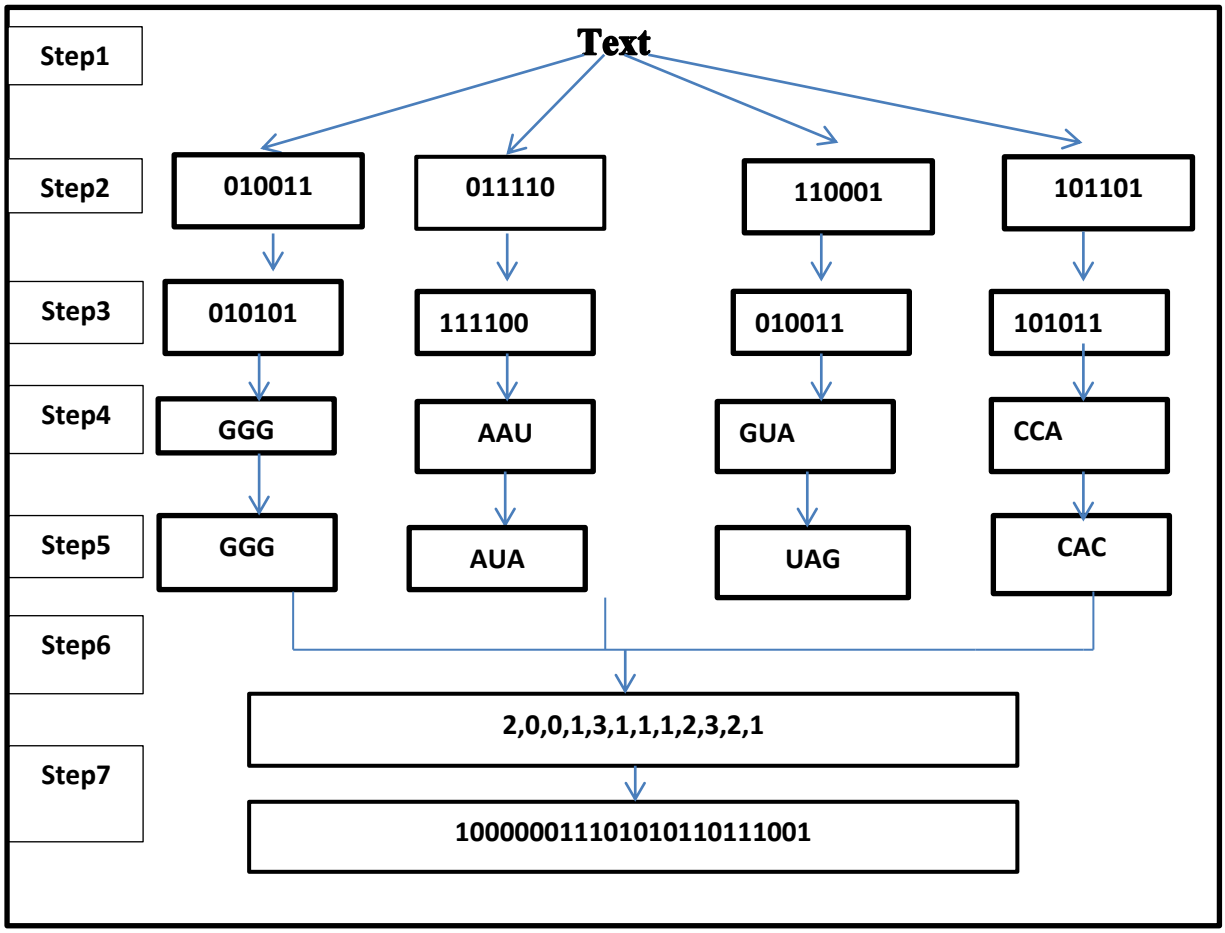

Figure 3- Example (1) of the proposed Method

First the text will be read which is according to the example is "Text" ,secondly the text characters will be converted to binary representation based private key1 ,thirdly every 6 bits will be permutated based private key2,fourthly the binary representation will be converted to RNA codes based private key3 ,fifthly the RNA codes will be permutated based private key4, sixthly the MTF will be applied on RNA codes to get decimal numbers, seventhly the decimal numbers will be converted to binary representation based private key5 , finally the binary representation will be sent.

Another example of the proposed method supposes that the text that we want to provide compression and encryption for it is "Computer". According to steps of the processing proposed method the text will be processed as follow in Figure-4.

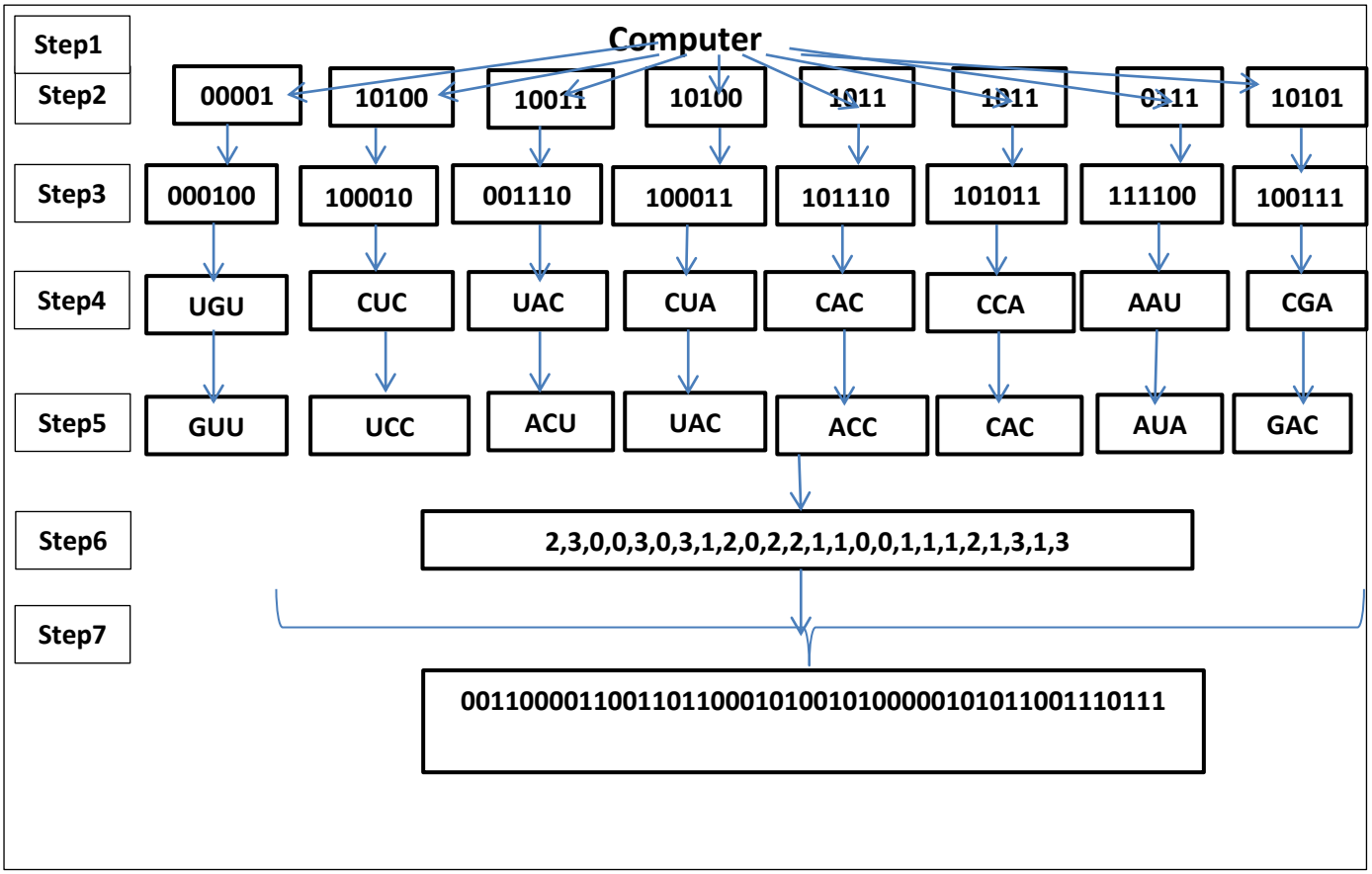

Figure 4- Example (2) of the proposed Method

First the text will be read which is according to the example is "Computer" ,secondly the text characters will be converted to binary representation based private key1 ,thirdly every 6 bits will be permutated based private key2,fourthly the binary representation will be converted to RNA codes based private key3 ,fifthly the RNA codes will be permutated based private key4, sixthly the MTFE will be applied on RNA codes to get decimal numbers, seventhly the decimal numbers will be converted to binary representation based private key5 , finally the binary representation will be sent.

6. Discussion of the proposed Method

The proposed method has the following features, every feature based on specific keys.

1) Security: The security of the proposed method is good based on the keys design since some of the keys have huge probabilities (private key1 has 64^{64} probabilities, private key2 has 6^6 probabilities, private key3 has 64^{64} probabilities where each element (6 bits) has 64 probabilities, private key4 has 3^3 probabilities, and private key5 has 2^2 probabilities), any change in the keys arrangement will provide a huge probabilities.

2) Compression: In normal cases every character has ASCII value and each value is (8 bits) in the proposed method every character will be represented in (6 bits) and processed according to the proposed method steps so the compression ratio for every character will be (6/8=0.75) and the compression factor will be (8/6=1.333).

7. Conclusions

The proposed method has good compression ratio and good security level. The compression level comes from using MTF method, since each 8 bits will be swapped with 6 bits. The good security level that the proposed method provides comes from based on multi private keys, RNA codes and MTF method together. The private keys are designed in especial ways which increase the security level. Every key has a lot of possibilities which increase the security level of the proposed method.

References

1. Sethi, G., Shaw, S., Vinutha, K. and Chakravorty, C. **2014**. Data Compression Techniques. *International Journal of Computer Science and Information Technologies*, **5** (4).
2. Mahajan, U., Prashanth, C.S.R. **2013**. Algorithms for Data Compression in Wireless Computing Systems. *IJCSI International Journal of Computer Science Issues*, **10**(5), No 1.
3. Kuppuswamy, P. and Alqahtani, Y. **2014**. New Innovation of Arabic Language Encryption Technique Using New Symmetric Key Algorithm. *International Journal of Advances in Engineering & Technology* , **7**(1), p30.
4. Harmouch, Y. and El Kouch, R. **2015**. A New Algorithm for Dynamic Encryption. *International Journal of Innovation and Applied Studies*, 10(1).
5. Singh, U. and Garg, U. **2013**. An ASCII value based text data encryption System. *International Journal of Scientific and Research Publications*, **3**(11).
6. Bazli, B., Tuncel, M.A. and Jones, D.L. **2014**. Data Encryption Using Bio Mlecular Information. *International Journal on Cryptography and Information Security (IJCIS)*, **4**(3).
7. Maqableh, M. M. **2012**. Analysis and Design Security Primitives Based on Chaotic Systems for e Commerce. Ph.D Thesis, Durham Unversity, School Of Engineering and Computing Sciences, United Kingdom.
8. Abdul Hassan, A.K. 2015. Proposed Approach for Key Generation Based on the RNA. *Journal of the College of basic education*, **21**(87).
9. Zalik, B. and Lukac, N. **2014**. Chain code lossless compression using move-to-front transform and adaptive run-length encoding. *International Journal of Signal processing and Image Communication*, **29**(1).