Proposed Data Coding Method

Dr. Suhad Malallah Kadhem
Computer Science Department, University of Technology/Baghdad
Email:suhad_malalla@yahoo.com
Sabrein Jassm
Computer Science Department, University of Technology/Baghdad
Email:SabreinJassm91@yahoo.com

Received on:10/3/2015 & Accepted on:17/12/2015

ABSTRACT

In data communications, a text is represented as a bit pattern ,a sequence of bits(0's or 1's).Different sets of bit patterns have been designed to represent text symbols .Each set is called a code ,and the process of representing symbols is called coding. In this paper a new coding method is proposed and in this method Move to front coding method is used. The result of applying the new coding method provides high complex code which can be employed for security purpose. Also look up table (s-boxes) is stored as a B^+ tree to provide efficient access and take less time in processing.

INTRODUCTION

Privacy is a major concern for users of public networks such as the Internet [1]. Encoding is one of the major concerns that are used to provide privacy [2]. There are many methods of encoding data, some of them are: open space methods that encode through manipulation of white space, syntactic methods that utilize punctuation, and semantic methods that encode using manipulation of the words themselves [3].In this Research a new coding method is proposed and in this method Move to front coding method and B^+ tree are used. The result of applying the new coding method provides high complex code which can be employed for security purpose.

The rest of this paper is organized as fallows section 2 will discuss coding operation, section3 will discuss data compression, section4 will discuss move to front coding method, section5 will discuss B^+ tree, section6 will discuss the proposed method with a given example for encoding and decoding, and the last section concludes this work.

Coding Operation

Coding theory refers to study of code properties and their suitability for specific applications. Efficient codes are used, e.g., data compression, cryptography, error-correction, steganography, and group testing. Codes play a central part in information theory, in particular in the design of efficient and reliable data transmission methods [4].

Data Compression

Data compression is the process of converting an input data stream in another data stream that is smaller in size .It is a key solution to the problem of huge storage required and long data

https://doi.org/10.30684/etj.34.2B.2
2412-0758/University of Technology-Iraq, Baghdad, Iraq

transmission requirements. The ratio of the original uncompressed file and the compressed file is referred to as the compression ratio [5]. The compression ratio is denoted by:

$$\text{Compression Ratio} = \frac{\text{UnCompresedFileSize}}{\text{CompressedFileSize}} = \frac{SIZE_U}{SIZE_C} \quad \ldots\ldots. (1).$$

It is often written as SIZEU: SIZEC.

Move To Front (MTF)

MTF transform is an encoding of data (typically a stream of bytes) designed to improve the performance of entropy encoding (coding scheme that assigns codes to symbols so (coding scheme that assigns codes to symbols so as to match code lengths with the probabilities of the symbols) techniques of compression. When properly implemented, it is fast enough that its benefits usually justify including it as an extra step in data compression algorithms [6].

The basic idea behind this technique is that there is a set A of alphabet, the indexing of this alphabet is used to make the compression. In addition, when making a compression to each character, its corresponding character is moved to the front of set A [6].

MTF transform can be illustrated in the following procedure. Consider the string that wants to be compressed is "ddbbadffcj" and alphabet is A= {a, b, c, d, f, j} the MTF steps will be as follow:

Set A=

A	B	C	D	F	J
0	1	2	3	4	5

1- Read the first character "D" the index = 3, and move the D to the front of the list

A=

D	A	B	C	F	J
0	1	2	3	4	5

index = 3

2- Read the second character "D" the index = 3,0 and D is already in the front of the list.

A=

D	A	B	C	F	J
0	1	2	3	4	5

index = 3,0

3- Read the next character "B" the index =3,0,2 and move "B" to the front of the list.

A=

B	D	A	C	F	J
0	1	2	3	4	5

index= 3,0,2

4- Read the next character "B" the index = 3,0,2,0 and B is already in the front of the list.

A=

B	D	A	C	F	J
0	1	2	3	4	5

index= 3,0,2,0

5- Read the next character "A" the index =3,0,2,0,2 and move "A" to the front of the list.

A	B	D	C	F	J

A=

0	1	2	3	4	5

index =3, 0,2,0,2

6- Read the next character "D" the index =3, 0,2,0,2, 2 and move "D" to the front of the list.

A=

D	A	B	C	F	J
0	1	2	3	4	5

index =3, 0,2,0,2, 2

and continue with the same procedure.

In decompressing the same procedure will be followed but instead of reading the text the compression code will be read and make a mapping to the set A.As illustrated in the following procedure.

Compression code is 3, 0,2,0,2,2

A=

A	B	C	D	F	J
0	1	2	3	4	5

Read the first number 3 and make a mapping to the set A, the text= D and move D to the front.

A=

D	A	B	C	F	J
0	1	2	3	4	5

Continue with the same procedure.

B^+ Tree

B^+ Tree is a variation of B-Trees a structure of nodes linked by pointers is anchored by a special node called the root, and bounded by leaves has a unique path to each leaf, and all paths are equal length stores keys only at leaves, and stores reference values in other, internal, nodes guides key search, via the reference values, from the root to the leaves. B+ tree is called an index to database, such that each record will be stored in the database, the reference number (and the key) of that record will be stored in the B+ tree. So when a certain record wants to be reached, its key needs to be known to get its reference number from the B^+ tree. When the reference number of that record has been gotten the required record can be retrieved directly. B^+ tree is an arranged and balanced tree, and this is why it is so fast in retrieving the required data. B^+ trees distinguish internal and leaf nodes, keeping data only at the leaves, whereas ordinary B -trees would also store keys in the interior. B^+ tree insertion, therefore, requires managing the interior node reference values in addition to simply finding a spot for the data, as in the simpler B-tree algorithm .B^+ tree is an arranged and balanced tree (see figure 1), and this is why it is so fast in retrieving the required data[7].

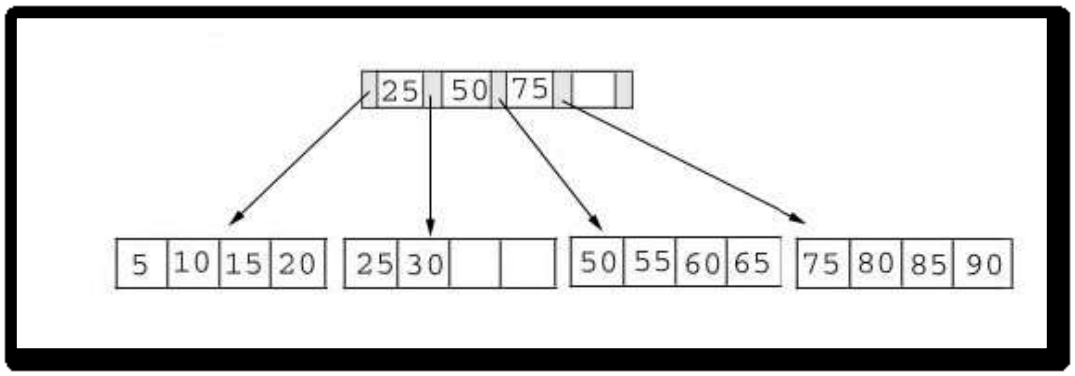


Figure (1) B^+ tree

The Proposed Method

Our method proposed that both sides (sender and receiver) have the following privet information:

- The number of applied Move to Front coding method.
- Move to front sets.
- Look up table (s-boxes) it's dimensions is (8*8) stored as B^+ tree to provide efficient access and take less time in processing.

The proposed method considered as coding method since it provides another form of the input which is the main feature of coding theory. Also the output form is simpler than the input form because the input is text (which is a sequence of characters) and produce compressed binary code. The proposed method also considered as compression method since it provides a good compression ratio since the proposed method replaces each 8 bits with 6 bits.

The proposed method also can be used for security purpose (as a cryptography method) because it takes text and converted it to another form which is unreadable form and the original text can't be retrieved just if the user has the above privet information. So to break this method the attacker needs 2^No.of.MTF (No.of.MTF is the number of applying MTF,2 because MTF is used in two stages before using S_box and after using S_box) probability which determine the number of used MTF sets , for example if we have No.of.MTF=5,so if the attacker know this number he will know the number of sets and every set take 2^{64} probabilities to know it's values($5*2^{64}$),so there a lot of strong point which make this algorithm preserved against attacks .

The following algorithm describes the main steps for the proposed method and we proposed that the number of using MTF method is 5 in the algorithm and the number(as well as it's sets) is variable can be increased or decreased according to the degree of the required security.

Proposed Coding Method Algorithm (Encoding)

Input: Arabic text**,** Set1, Set2, Set3, Set4, Set5, S-box1 its dimension 8*8 contains decimal values from 0-64; S-box2 its dimension 8*8 contains values from 0-64.

Output: Binary code.

Begin

Step1: Read Text.

Step2: Split the text into separated characters and put them into list1 (which is a list of characters).

Step3: Apply MTF coding method by using set1 and get list2 (which is a list of integer values).

Step4: Consider each value of list2 as ASCII number, get the corresponding character and put result in list3 (which is a list of characters).

Step5: Apply MTF coding method to list3 by using set2, to get list4 (which is a list of integer values).
Step6: For each value of list4, do the following:
Step6.1: Find the binary representation for each six bits.
Step6.2: Take the first three bits; convert them to decimal to be considered as the row of s-box.
Step6.3: Take the last three bits; convert them to decimal to be considered as the column of s-box.
Step6.4: Find the index value of the specified row and column in s-box1.
Step6.5: Take the value of previous step and consider it as row and column in s-box2.
Step6.6: Find the value of the specified row and column in s-box2.
Step6.7: Put the result in list5.
Step7: Apply move to front to list5 using set3, get list6.
Step8: Apply move to front to list6 using set4, get list7.
Step9: Apply move to front to list7 using set5, get list8.
Step10: For each member of list8 convert to binary representation and put in list9.
Step11: Send Binary code.
End

Proposed Coding Method Algorithm (Decoding)
Input: Binary code**,** Set1, Set2, Set3, Set4, Set5, S-box1 its dimension 8*8 contains decimal values from 0-64, S-box2 it's dimension 8*8 contains values from 0-64.
Output: Arabic text.
Begin
Step1: Receive Binary code.
Step2: Take each 6 bits; convert to decimal representation, put in list1.
Step3: Apply move to front decoding to list1 using set5 and get list2.
Step4: Apply move to front decoding to list2 using set4 and get list3.
Step5: Apply move to front decoding to list3 using set3 and get list4.
Step6: For each value of list4, do the following:
Step6.1: Take the Value and look for it in s-box2, return it row.
Step6.2: Take the row value, consider it as value and finds its row and column in s- Box1.
Step6.3: Take the row and column, return their binary representation, store them in B.
Step6.4: Find corresponding decimal value for B, put the result in list5.
Step7: Apply move to front decoding to list5 using set2 and get list6.
Step8: For each character in list6 find its ASCII, put in list7.
Step9: Apply move to front decoding to list7 using set1 and get list8.
Step10: Return the text.
End

Result

A. Encoding Phase

1) Read Text.
العراق وطني
2) Split the text into separated characters and put them into list1 (which is list of characters).
List1= [ا ل ع ر ا ق و ط ن ي]
3) Apply move to front coding using set1 and get list2 (which is list of integer values).
List2= [٢٨ ١٦ ١٨ ١٩ ٧ ٣ ٢١ ٠ ٢ ١٢]

4) Consider each value of list2 as ASCII number, get the corresponding character and put result in list3 (which is of characters).

List3= [و ج ز ق م ا خ ف ش]

5) Apply move to front coding to list3 using set2, get list4 (which is list of integer values).

List4= [٤ ٦ ١ ٢٠ ١٧ ١٣ ٦٢ ١٢ ٢٧ ٢٣]

6) For each value of list4, do the following: find binary representation (6 bits), take the first 3 bits, convert to decimal and consider it as row of s-box, take the last 3 bits, convert to decimal and consider it as column of s-box, find the value of the specified row and column in s-box1, find the value of the specified row and column in s-box2,and put the result in list5.

For example we will take each decimal element in list4 (e.g. 23) convert to binary representation(010111) take the first 3 bits($(010)_2$-----$(2)_{10}$) , considered as row in s_box table and take the last 3 bits((111)2-----(7)10), considered as column in s_box table ,then take the value of the intersection of these row and column and insert it in list5(e.g.39).This value is replacement of 23,this process will continue until all list5 is complete.

After applying swapping with S_Box values the result will be like the values in List5

List5= [٥٨ ٥٦ ٦١ ٤٢ ٤٥ ٤٩ ٠ ٥٠ ٣٥ ٣٩]

7) Apply move to front to list5 using set3, get list6.

List6= [٤٣ ٤٥ ٦١ ٥٩ ٥٦ ٥٢ ٢٠ ٥١ ٢٦ ٢٢]

8) Apply move to front to list6 using set4, get list7.

List7= [٥١ ٤٩ ٣١ ٣٥ ٣٨ ٤٢ ١٠ ٤٣ ٤ ٨]

9) Apply move to front to list7 using set5, get list8.

List8= [١٢ ١٤ ٣٢ ٢٨ ٢٥ ٢١ ٥٣ ٢٠ ٥٩ ٥٥]

10) For each member of list8 convert to binary representation and put in list9.

List9= [٠٠١١٠٠ ٠٠١١١٠ ١٠٠٠٠٠ ٠١١١٠٠ ٠١١٠٠١ ٠١٠١٠١ ١١٠١٠١ ٠١٠١٠٠ ١١١٠١١ ١١٠١١١].

11) Send Binary code.

Code=٠٠١١٠٠٠٠١١١٠١٠٠٠٠٠١١١٠٠٠١١٠٠١٠١٠١٠١١١٠١٠١٠١٠١٠٠١١١٠١١١١٠١١

B. Decoding Phase

1) Receive Binary code.

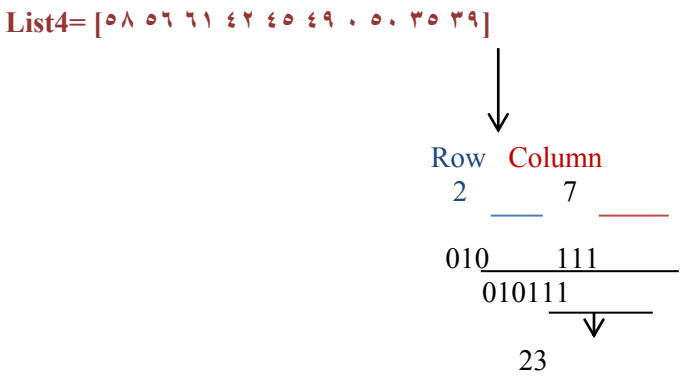
Code=٠٠١١٠٠٠٠١١١٠١٠٠٠٠٠١١١٠٠٠١١٠٠١٠١٠١٠١١١٠١٠١٠١٠١٠٠١١١٠١١١١٠١١١

2) Take each 6 bits (list0), convert to decimal representation, put in list2.

List0=[٠٠١١٠٠,٠٠١١١٠,١٠٠٠٠٠,٠١١١٠٠,٠١١٠٠١,٠١٠١٠١,١١٠١٠١,٠١٠١٠٠,١١١٠١١,١١٠١١١].

List1= [١٢ ١٤ ٣٢ ٢٨ ٥٢ ٢١ ٥٣ ٢٠ ٥٩ ٥٥]

3) Apply move to front decoding to list1 using set5 and get list2.
List2= [٥١ ٤٩ ٣١ ٣٥ ٣٨ ٤٢ ١٠ ٤٣ ٤ ٨]
4) Apply move to front decoding to list2 using set4 and get list3.
List3= [٤٣ ٤٥ ٦١ ٥٩ ٥٦ ٥٢ ٢٠ ٥١ ٢٦ ٢٢]
5) Apply move to front decoding to list3 using set3 and get list4.
List4= [٥٨ ٥٦ ٦١ ٤٢ ٤٥ ٤٩ ٠ ٥٠ ٣٥ ٣٩]
5) For each value of list4, do the following: take the Value and look for it in s-box, return it's row and column .Take the row and column, return their binary representation and combining their binary representation together ,store result in B. Find corresponding decimal value for B, put the result in list5.



For example we will take each value in List4 ,(e.g.39) look for this value in s_box table and return it's row(2) and column(7) .These row and column convert to binary representation(Row=010,Column=111) ,combine these binary values together, get the binary representation of 6 bits(010111) ,and get the decimal value corresponding to this binary representation(23),insert it in list5,this process will continue until the list5 complete.
List5= [٤ ٦ ١ ٢٠ ١٧ ١٣ ٦٢ ١٢ ٢٧ ٢٣]
6) Apply move to front decoding to list5 using set2 and get list6.
List6= [و ج ز ق م ا خ ف ش]
7) For each character in list6 find its ASCII, put in list7.
List7=[٢٨ ١٦ ١٨ ١٩ ٧ ٣ ٢١ ٠ ٢ ١٢]
8) Apply move to front decoding to list7 using set1 and get list8.
List8=[ا ل ع ر ا ق و ط ن ي]
9) Return the text.
العراق وطني

CONCLUSIONS

1-The new proposed method provides high complexity and security (according to the number of times the MTF is applied) for coding and decoding operations, so it is considered as a new proposed security method or it can be a step in an application that required high security.
2-Give a good compression ratio since the proposed method replaces each 8 bits with 6 bits.

3-Represent look up table (s-boxes) by using B^+ tree provides efficient access and takes less time in processing.

Appendix

Table (1) :- S-box

Column / Row	0	1	2	3	4	5	6	7
0	63	61	60	59	58	57	56	55
1	54	53	52	51	50	49	48	47
2	46	45	44	43	42	41	40	39
3	38	37	36	35	34	33	32	31
4	30	29	28	27	26	25	24	23
5	1	42	11	20	18	17	16	15
6	48	49	10	12	10	9	8	7
7	6	5	4	3	2	1	0	62

Table (2):- Set (1)

ح	خ	د	س	ش	غ	ف	ك	ع	ذ
0	1	2	3	4	5	6	7	8	9

ز	ء	ا	خ	ظ	ف	ن	ه	ط	و
10	11	12	13	14	15	16	17	18	19

ح	ر	ب	ش	ت	ث	د	ذ	ي	ُ
20	21	22	23	24	25	26	27	28	29

◌َ	◌ِ	◌ُ	◌ٍ	◌ً	١	٢	٣	٤	٥
30	31	32	33	34	35	36	37	38	39

٦	٧	٨	٩	٠	◌ْ	ة	~	آ	أ

40	41	42	43	44	45	46	47	48	49

ئ	ؤ	إ	:	؟	"	،	.	ى	!
50	51	52	53	54	55	56	57	58	59

٠	*	%	*
60	61	62	63

Table (3):- Set (٢)

ح	خ	د	س	ش	غ	ف	ك	ع	ذ
0	1	2	3	4	5	6	7	8	9

ز	ق	ص	ظ	ل	م	ط	ن	ا	ء
10	11	12	13	14	15	16	17	18	19

ت	و	ي	ب	ث	ج	٠	%	*	;
20	21	22	23	24	25	26	27	28	29

،	!	ى	.	◌َ	◌ِ	؟	:	إ	ؤ
30	31	32	33	34	35	36	37	38	39

ئ	ة	ر	أ	آ	~	◌ْ	٩	٨	٧
40	41	42	43	44	45	46	47	48	49

٦	٥	٤	٣	٢	١	◌ُ	◌ّ	◌ً	-
50	51	52	53	54	55	56	57	58	59

“	ض	#	‘
60	61	62	63

Table (4):- Set (3)

٢٠	١٩	١٨	١٧	١٦	١٥	١٤	١٣	١٢	١١
0	1	2	3	4	5	6	7	8	9

١٠	٩	٨	٧	٦	٥	٤	٣	٢	١
10	11	12	13	14	15	16	17	18	19

٠	٤٠	٣٩	٣٨	٣٧	٣٦	٣٥	٣٤	٣٣	٣٢
20	21	22	23	24	25	26	27	28	29

٣١	٣٠	٢٩	٢٨	٢٧	٢٦	٢٥	٢٤	٢٣	٢٢
30	31	32	33	34	35	36	37	38	39

٢١	٦٠	٥٩	٥٨	٥٧	٥٦	٥٥	٥٤	٥٣	٥٢
40	41	42	43	44	45	46	47	48	49

٥١	٥٠	٤٩	٤٨	٤٧	٤٦	٤٥	٤٤	٤٣	٤٢
50	51	52	53	54	55	56	57	58	59

٤١	٦٣	٦٢	٦١
60	61	62	63

Table (5):- Set (٤)

٣٠	٢٩	٢٨	٢٧	٢٦	٢٥	٢٤	٢٣	٢٢	٢١
0	1	2	3	4	5	6	7	8	9

٢٠	١٩	١٨	١٧	١٦	١٥	١٤	١٣	١٢	١١
10	11	12	13	14	15	16	17	18	19

١٠	٩	٨	٧	٦	٥	٤	٣	٢	١
20	21	22	23	24	25	26	27	28	29

٠	٦٣	٦٢	٦١	٦٠	٥٩	٥٨	٥٧	٥٦	٥٥
30	31	32	33	34	35	36	37	38	39

٥٤	٥٣	٥٢	٥١	٥٠	٤٩	٤٨	٤٧	٤٦	٤٥
40	41	42	43	44	45	46	47	48	49

٤٤	٤٣	٤٢	٤١	٤٠	٣٩	٣٨	٣٧	٣٦	٣٥
50	51	52	53	54	55	56	57	58	59

٣٤	٣٣	٣٢	٣١
60	61	62	63

Table (6):- Set (٥)

٦٣	٦٢	٦١	٦٠	٥٩	٥٨	٥٧	٥٦	٥٥	٥٤
0	1	2	3	4	5	6	7	8	9

٥٣	٥٢	٥١	٥٠	٤٩	٤٨	٤٧	٤٦	٤٥	٤٤
10	11	12	13	14	15	16	17	18	19

٤٣	٤٢	٤١	٤٠	٣٩	٣٨	٣٧	٣٦	٣٥	٣٤
20	21	22	23	24	25	26	27	28	29

٣٣	٣٢	٣١	٣٠	٢٩	٢٨	٢٧	٢٦	٢٥	٢٤
30	31	32	33	34	35	36	37	38	39

٢٣	٢٢	٢١	٢٠	١٩	١٨	١٧	١٦	١٥	١٤
40	41	42	43	44	45	46	47	48	49

١٣	١٢	١١	١٠	٩	٨	٧	٦	٥	٤
50	51	52	53	54	55	56	57	58	59

٣	٢	١	٠
60	61	62	63

REFERENCES

[1] Akbas E.A., "A New Text Steganography Method By Using Non Printing Unicode Characters", Eng & Tech Journal, Vol.28, No.1, 2010.

[2] Behrouz A.Forouzan,"Data Communications And Networking", 2007.

[3] Auday Jamal Fawzi ,"Data Hiding in Arabic Text", PhD thesis , University of Technology , Iraq ,January,2007 .

[4] James Irvine, David Harle, "Data Communications and Networks". 2002.

[5] Nidaa F.H., "Robust Method For Hiding Text In Wave File Against MP3 Compression" , Ph.D. thesis, Department of Computer Science ,The University of Technology, 2005.

[6] David Salomon, Data Compression- The Complete Reference, Third Edition, 2004.

[7] Suhad M. Kadhem," Using B+ Tree To Represent Secret Messages For Steganography Purpose", Eng & Tech Journal, Vol (28),No(15),Iraq , March 2010.