Stochastic Modelling and Computational Sciences
Vol. 1 No. 1 (June, 2021)

Received: 15th January 2021 *Revised: 21st March 2021* *Accepted: 10th May 2021*

H2O Algorithm Theorem for Secure Communication System

H.K. Hu, Chung-Ang University, Seoul, Korea

Abstract

In this paper we have developed and designed H_2O (**H**ackers have **H**urdle in breaking the **O**mega network) algorithm for encryption and decryption. Cryptography provides many methods and techniques for secure communication. Apart from the existence of many techniques, we proposed a new technique. To generate the key, we used bit-level encryption methodologies like substitution, permutation [5].On focusing the security, along with Shuffle-Exchange (Omega network) [3] the conversion of binary to gray code is done. The key length is 592 bit which is hard to break. These factors strengthen the algorithm. Since the block cipher technique is used, the speed of the algorithm is increased. This paper discusses on how to generate encryption key from which substitution table is constructed and the information is enciphered.

Keywords: Cryptography, Encryption, Decryption, Omega network, Substitution, Permutation

1. INTRODUCTION

Security is the primary concern of all those people who deal with activities, which involve protection of risk. The branch of science “cryptography” is concerned with the security of information, developed in the hands of military people and it was nurtured by them for quite long time as their private property. For this reason many algorithms are developed for encryption and decryption which provide high security. All these algorithms are kept open to the public and the secrecy of the algorithm lies entirely in the key. This paper stands different that the development of our own algorithm addresses the user needs in specific, there by more flexibility.

2. LITERATURE REVIEW

2.1 Problem Definition

In this we have taken plain text as a text file. This file will have all the ASCII characters. The ASCII characters are given in the Table 2.1. In this table the rows

and column indicates the left digit and the right digit of the ASCII value of the corresponding character

For eg. Consider the row with numbered **3** towards the combination of column numbered **3** represents **33** which is the ASCII value of **!**.

Table 2.1
ASCII Table

	0	1	2	3	4	5	6	7	8	9
3			Blank	!	“	#	$	%	&	‘
4	(	)	*	+	,	-	.	/	0	1
5	2	3	4	5	6	7	8	9	:	;
6	<	=	>	?	@	A	B	C	D	E
7	F	G	H	I	J	K	L	M	N	O
8	P	Q	R	S	T	U	V	W	X	Y
9	Z	[	\	]	^	_	‘	a	b	c
10	d	E	F	g	h	i	j	k	l	m
11	n	O	P	q	r	s	t	u	v	w
12	x	Y	Z	{	|	}	~			

This table is divided into 32 blocks, starting from 1 to 31 and finally 0 such that all the blocks contain 3 characters except the 20th block which contain 2 characters. This is represented in Table 2.2

Table 2.2

Block Number	*Character*	*Number of Character*	*Block Number*	*Character*	*Number of Character*
01	Blank ! “	3	17	P Q R	3
02	# $ %	3	18	S R U	3
03	& ‘ (	3	19	V W X	3
04	) * +	3	20	Y Z	2
05	, - .	3	21	[\ ]	3
06	/ 0 1	3	22	^ _ ‘	3
07	2 3 4	3	23	a b c	3
08	5 6 7	3	24	d e f	3
09	8 9 :	3	25	g h i	3
10	; < =	3	26	j k l	3
11	> ? @	3	27	m n o	3
12	A B C	3	28	p q r	3
13	D E F	3	29	s t u	3
14	G H I	3	30	v w x	3
15	J K L	3	31	y z {	3
16	M N O	3	00	| } ~	3

For developing the encryption and decryption algorithm, we take a typical key $\mathbf{K_0}$ as follows

K_0	=	00	30	28	26	24	22	20	18	16	14	12	10
		08	06	04	02	01	03	05	07	09	11	13	15
		17	19	21	23	05	27	29	31				

[Block Number 0 to 31 occurring in the key K_0 corresponding to the Table 2.2, arranged randomly]

3. IMPLEMENTATION

Algorithm for Encryption & Decryption

Encryption:

1. Take the key $\mathbf{K_0}$
2. Permute the key $\mathbf{K_0}$ using Permuted key $\mathbf{K_P}$ and we get Encryption Key, $\mathbf{K_E}^*$
3. Construct the substitution table based upon $\mathbf{K_E}^*$
4. Read the plaintext from the file.
5. Obtain the decimal number corresponding to the character by using the substitution table, thus we get Substituted text.

Substituted text + Encryption Key = Encrypted Text

6. Shuffle the Encrypted text.
7. Convert the Binary coded information to Gray coded information.

Decryption

1. Construct the Substitution table using the Encryption Key received.
2. Convert the Gray code Encrypted information to Binary code Encrypted information.
3. Shuffle the information to get the actual information as before shuffling in the Encryption Algorithm.
4. Obtain the Decimal equivalent of the binary.
5. Search for the decimal number in the **substitution table**, and get the Row and Column value which is corresponding to the ASCII value of the Encrypted character.

3.1. Explanation for Encryption:

Step 1:

Take the Key $K_{0.}$

Step 2:

Permutation is carried out by interchanging bits by using Permuted key $\mathbf{K_P}$ (1423451235) in K_0. This Permuted key also transmitted to receiver in the secured manner. Thus, we get the Encryption key $\mathbf{K_E^*}$

K_0	=	00	30	28	26	24	22	20	18	16	14	12	10
		08	06	04	02	01	03	05	07	09	11	13	15
		17	19	21	23	05	27	29	31				

Eg. The binary equivalent for **30 is** **1 1 1 1 0**

Bit No: 1 2 3 4 5

=> Swapping 1st and 4th bit

1 1 1 1 0

1 2 3 4 5

=> Swapping 2nd and 3rd bit

1 1 1 1 0

1 2 3 4 5

=> Swapping 4th and 5th bit

1 1 1 0 1

1 2 3 4 5

=>Swapping 1st and 2nd bit

1 1 1 0 1

1 2 3 4 5

=>Swapping 3rd and 5th bit

1 1 1 0 1

1 2 3 4 5

The decimal Equivalent of **1 1 1 0 1 is 29**

For the rest of the Numbers in K_0, the similar procedure is implemented, thus we get $\mathbf{K_E^*}$ as below.

K_E^*	=	00	29	21	13	05	28	20	12	04	25	17	09
		01	24	16	08	02	10	18	26	03	11	19	27
		06	14	22	30	07	15	23	31				

Step 3:

Using the Encryption key, we have to construct the substitution table. Consider the first number in $\mathbf{K_E^*}$, that is 00 and it refers to the block 00 in Table 2. This block contains these characters | } and ~, the corresponding ASCII value (124,125,126)

of these character is obtained from the Table 1 and replace these characters by decimal 0,1,2 in ASCII Table thus forms the Substitution Table.

This procedure is repeated for the remaining Numbers in $\mathbf{K_E}^*$

Table 3.1
Substitution Table

	0	1	2	3	4	5	6	7	8	9
3			35	36	37	47	48	49	59	60
4	61	23	24	25	12	13	14	71	72	73
5	83	84	85	44	45	46	32	33	34	50
6	51	52	62	63	64	20	21	22	9	10
7	11	74	75	76	86	87	88	41	42	43
8	29	30	31	53	54	55	65	66	67	18
9	19	6	7	8	77	78	79	89	90	91
10	38	39	40	26	27	28	56	57	58	68
11	69	70	15	16	17	3	4	5	80	81
12	82	92	93	94	0	1	2			

The Key K_E is transmitted to the receiver by the following procedure / method,

Considering the key as a Double digit Number,

Key K_E = K_0 + K_P where K_P (14, 23, 45, 12, 35)

= 32 + 5

= 37 Double digit Key

Considering the Key as a Single digit Number,

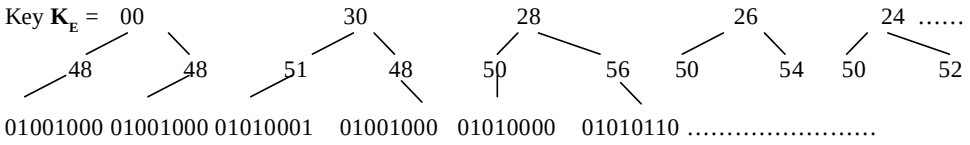
Key K_E = K_0 + K_P where K_P (1, 4, 2, 3, 4, 5, 1, 2, 3, 5)

= 64 + 10

= 74 Single digit Key

Now we are representing each digit K_E by its corresponding ASCII value, thus we will get 148 (74 * 2) digit of key for eg. Consider the key 30 in K_0 in which the ASCII value of 3 and 0 is 51 and 48 respectively thus 30⇔5148. The next step is to represent each digit in 4 bit binary form. Thus the key length in our work is 592 bit (148 * 4), so it is not that much easy for the Hacker to break the key. Similarly it is repeated for the entire Key element in KE.

The above procedure is illustrated as follows,

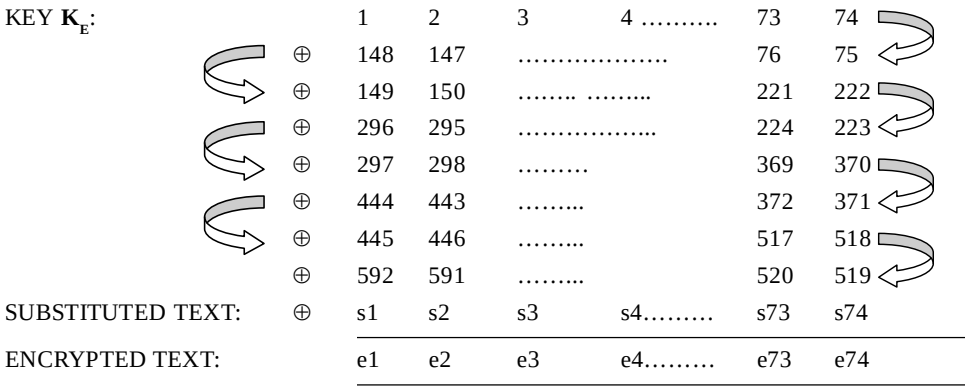


Step 4

Read the text

Step 5

"Encrypted Text = Substituted Text + Key $\mathbf{K_E}$", we are going to take 74 bit of Substituted text and dividing the key of 592 bits into 8 parts thus we get eight 74 bits of key, and with "Exclusive OR" operation along with the Substituted text we will get 74 bit of Encrypted text/information which is pictorially represented below.



1, 2, 3……592 represents bit numbers of key K_E

s1, s2, s3…….s74 represents bit numbers of substituted text

e1, e2, e3…….e74 represents bit numbers of encrypted text

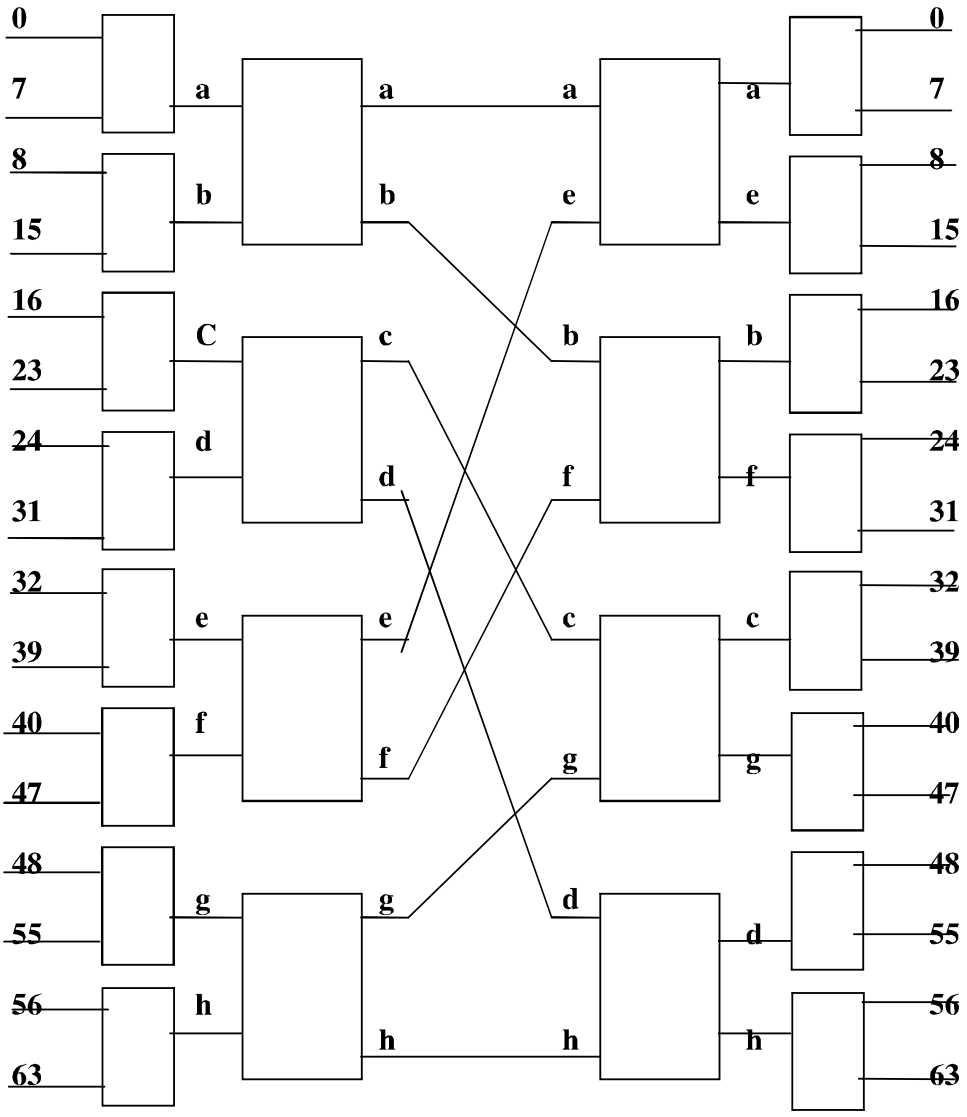
⊕ - Exclusive OR

Step 6

Shuffle Network for Encryption

The shuffle network for Encryption is given below. Here we are taking 64 bit of Encrypted text and converting each 8 bit into Decimal equivalent and fed them

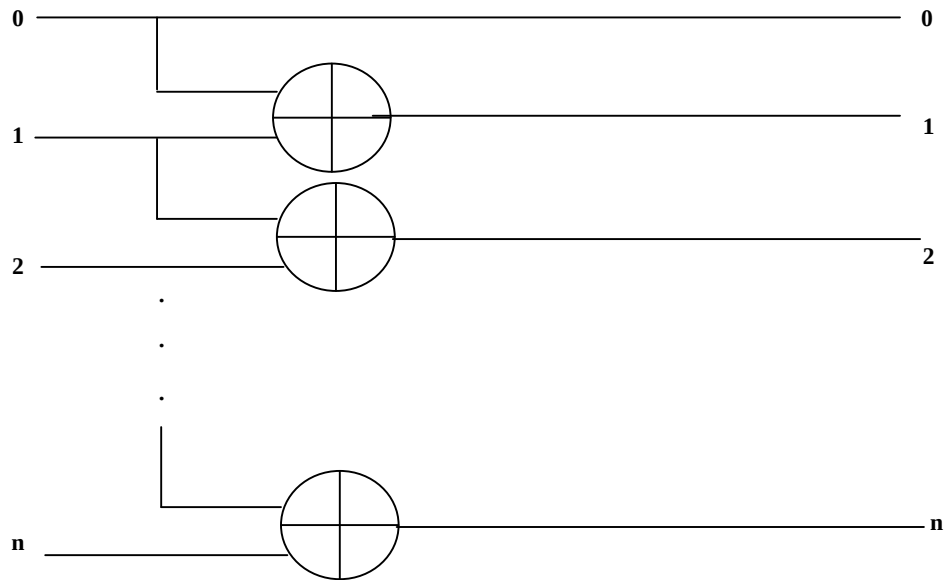
through the Exchange box where the Decimal value get swapped. After exchanging we are converting the Decimal to its Binary equivalent.



B/D	**-**	**Binary to Decimal Converter**	**0, 1, 2……63**	**-**	**Binary bit**
E	**-**	**Exchange Box**	**a, b, c…h**	**-**	**Decimal Digit**
D/B	**-**	**Decimal to Binary Converter**			

Step 7

Convert the Binary coded Encrypted Information to Gray coded Encrypted Information



3.2 Explanation for Decryption

Step 1

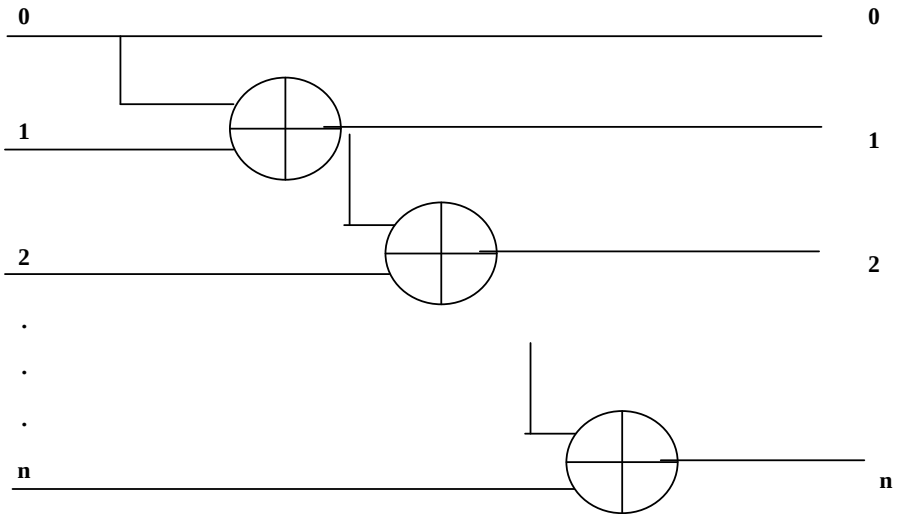
Construction of Substitution Table from Key

1. Convert the 592 bit of key K_E to its decimal value, split it into 148 blocks such that each block contains 4 bit of binary.
2. Represent the 4 bit of binary into corresponding Decimal (ie. we are having 148 single decimal digit).
3. Combine two Decimal digits to form single double digit Number. (ie. 74 double digit Numbers).
4. This 74 Double digit Number reflects the corresponding ASCII value of Key $K_{E.}$ (74 Single Digit number).
5. We obtain 37 Double digit numbers by combining the above single digit Number. Among 37 Double digits Number the first 32 Number reflects the Key (K_0) and the rest for Swapping purpose.

6. We obtain the Encryption key (K_E^*) from K_0 by applying the Permutation procedure.
7. Construct the Substitution Table using K_E^*.

Step 2

Convert Gray coded Encrypted Information to Binary coded Encrypted Information:



Step 3

Shuffle Network for Decryption

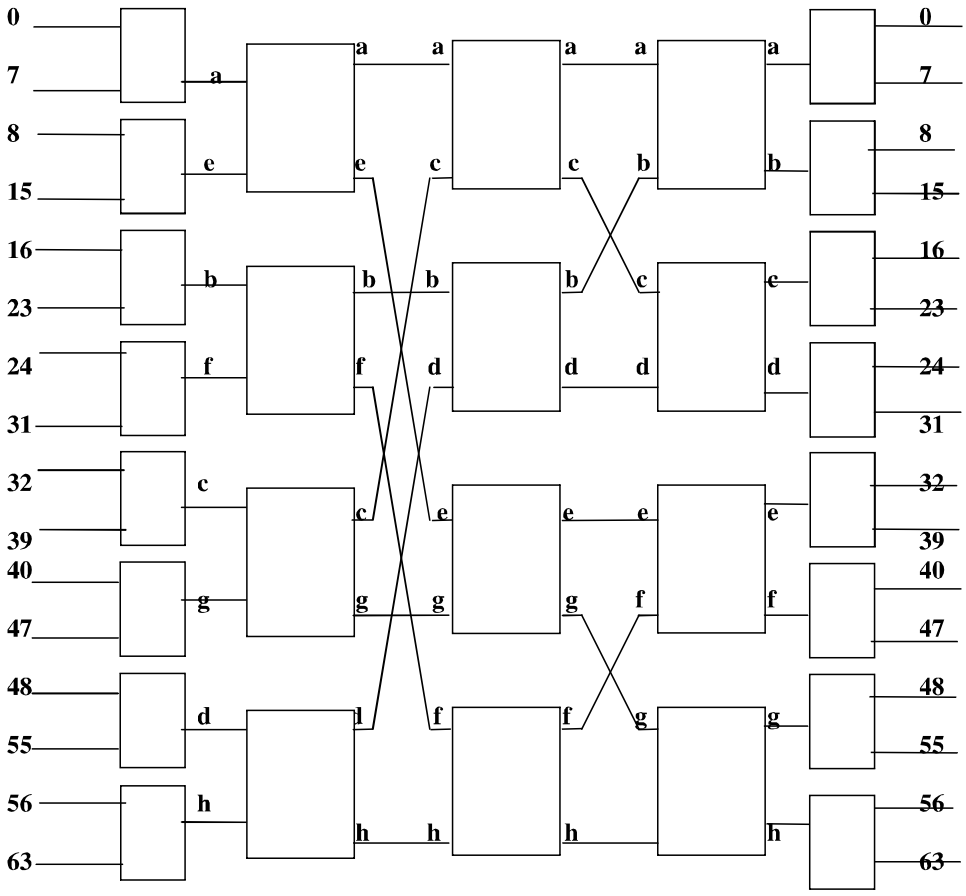
As we have shuffled the information in the Encryption side, it is necessary for the receiver to again shuffle the information to get the original information. The process is very similar to the Shuffle network for Encryption, the only thing is that we are shuffling to get the original information.

Step 4

Obtain the Decimal equivalent of the Binary.

Step 5

Search for the Decimal number in the substitution table, and get the row and column value which is corresponding to the ASCII value of the Encrypted character.



B/D - Binary to Decimal Converter
E - Exchange box
D/B - Decimal to Binary Converter

0, 1, 2……63 - Binary bit
a, b, c….h - Decimal Digit

4. CONCLUSION

This technique is scarcely impossible to break the code .As the key length is 592, it takes 592! combinations which is not imaginable and so it is not easy to guess the exact key combination. High level security is achieved by this algorithm.

REFERENCES

[1] Bruce Schenier, “Applied Cryptography”.

[2] Kai Hwang, “Advanced Computer Architecture”.

[3] Kai Hwang, Faye A.Briggs, “Computer Architecture and Parallel Processing”, McGraw Hill Edn.

[4] Michel J. Quion, “Parallel Computation”, McGraw Hill Edn

[5] William Stallings, “Cryptography and Network Security”, Pearson Education.

[6] http://www.wikipedia.com