Volume 3, Issue 7, July 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

17- Level Enciphering Algorithm for Enhanced Security

Sachi Jain
Department of Computer Engineering,
I.E.T DAVV, INDORE, India

Abhinav Shukla
Department of Electronics and Instrumentation
I.E.T DAVV, INDORE, India

Abstract— Due to the rise in technology, it has become easier to hack confidential data. At the same time if that sensitive data or information is presented in an encrypted form, would increase the needed confidentiality. Without encryption there is no such thing like privacy. Considering the computing power of nowadays computers, enciphering will require a very less time. The algorithm presented in this paper would efficiently encrypt the data..

Keywords— ASCII, REPEATVAL, Encipher

I. INTRODUCTION

An enciphering algorithm is a computational procedure used to encrypt data. Security algorithms are effective tools to secure information from unauthenticated access. These are the set of instructions to prevent fraudulent intrusion. Here in the proposed algorithm, user given input is transformed into a complicated combination of digits. A design combining 7 modules in a flow is explained below.

II. MODULES

A. *Module-1*

Take the input and count the length of the input string.

1) At every odd position of the string, replace the character with a new character whose ASCII value is equal to the ASCII value of the original character + length of the string. (If the ASCII value exceeds 127, acquire the modulo of the exceeding value, this will give an ASCII value for the corresponding character.)(Refer to the ASCII table below.)

2) Divide the length of the string by 3 and shift the string to the right by value of quotient irrespective of the remainder.

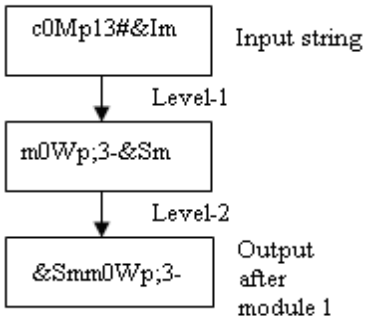

Fig. 1 Module-1

B. *Module-2*

3) Count the number of repeating characters and save the value in "REPEATVAL". With repeating characters, replace the first occurrence of that character with the position of the second occurrence.

4) Attach a character to beginning obtained by character whose ASCII value = 33 + REPEATVAL. (If exceeds 127, do the same as in step 1.)

5) Append the string with the character value whose ASCII value = 33 + 2*REPEATVAL. (If exceeds 127, do the same as in step 1.)

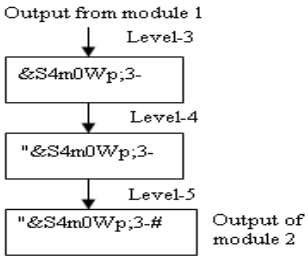

Fig. 2 Module-2

C. Module-3

6) Shift the symbols (except alphabets and numbers) right by a value = quotient of (number of symbols divided by 2).
7) After every symbol attach in number the position of that symbol.
8) Change al alphabets from lower case to upper case and vice versa.

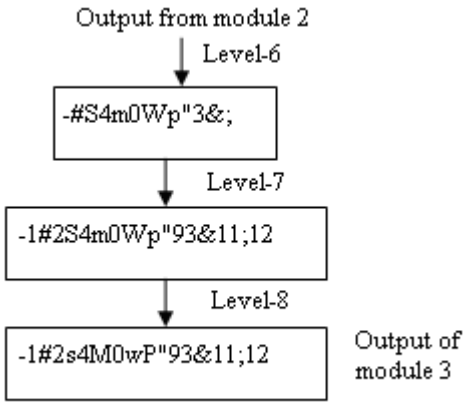

Fig. 3 Module 3

D. Module-4

9) Collect the symbols (characters) at all the odd positions in set A and all at the even ones in set B.
10) Further collect the characters at all the odd positions from set A in set P and at the even ones from set A into set Q. Do the same process in set B to collect odd position characters in set R and even ones in set S.
11) Append set R to set P and set Q to set S, to get new sets set W and set X.

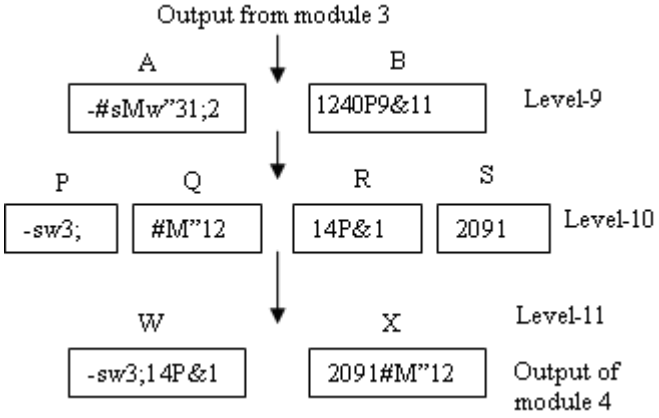

Fig. 4 Module 4

E. Module-5

12) Reverse set W and append it to set X.
13) If there are two consecutive digits replace them with the corresponding symbol in the ASCII table below.
14) Convert each alphabet to its ASCII value.

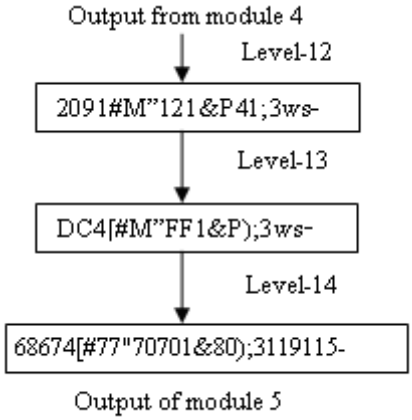

Fig. 5 Module 5

F. Module-6

15) Swap characters at position 1 with 2, 3 with 4 and so on. Leave the last one as it as if not found in a pair.
16) Shift the symbols (except numbers) right by the value equal to the first non-zero number present in the string.
17) replace all symbols (except numbers) with the last digit of there ASCII equivalent.

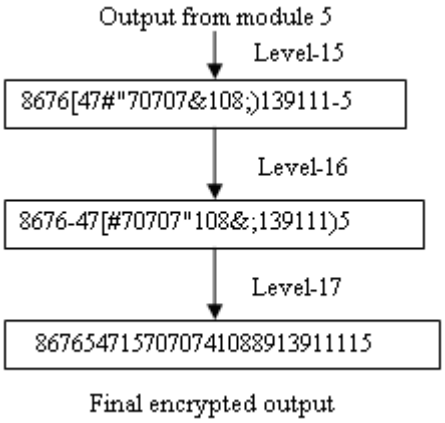

Fig. 6 Module 6

III. ASCII TABLE

The ASCII conversion table given here would be referred in certain modules as needed.

ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol
0	0	NUL	16	10	DLE	32	20	(space)	48	30	0
1	1	SOH	17	11	DC1	33	21	!	49	31	1
2	2	STX	18	12	DC2	34	22	"	50	32	2
3	3	ETX	19	13	DC3	35	23	#	51	33	3
4	4	EOT	20	14	DC4	36	24	$	52	34	4
5	5	ENQ	21	15	NAK	37	25	%	53	35	5
6	6	ACK	22	16	SYN	38	26	&	54	36	6
7	7	BEL	23	17	ETB	39	27	'	55	37	7
8	8	BS	24	18	CAN	40	28	(	56	38	8
9	9	TAB	25	19	EM	41	29	)	57	39	9
10	A	LF	26	1A	SUB	42	2A	*	58	3A	:
11	B	VT	27	1B	ESC	43	2B	+	59	3B	;
12	C	FF	28	1C	FS	44	2C	,	60	3C	<
13	D	CR	29	1D	GS	45	2D	-	61	3D	=
14	E	SO	30	1E	RS	46	2E	.	62	3E	>
15	F	SI	31	1F	US	47	2F	/	63	3F	?

ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol
64	40	@	80	50	P	96	60	`	112	70	p
65	41	A	81	51	Q	97	61	a	113	71	q
66	42	B	82	52	R	98	62	b	114	72	r
67	43	C	83	53	S	99	63	c	115	73	s
68	44	D	84	54	T	100	64	d	116	74	t
69	45	E	85	55	U	101	65	e	117	75	u
70	46	F	86	56	V	102	66	f	118	76	v
71	47	G	87	57	W	103	67	g	119	77	w
72	48	H	88	58	X	104	68	h	120	78	x
73	49	I	89	59	Y	105	69	i	121	79	y
74	4A	J	90	5A	Z	106	6A	j	122	7A	z
75	4B	K	91	5B	[	107	6B	k	123	7B	{
76	4C	L	92	5C	\	108	6C	l	124	7C	|
77	4D	M	93	5D	]	109	6D	m	125	7D	}
78	4E	N	94	5E	^	110	6E	n	126	7E	~
79	4F	O	95	5F	_	111	6F	o	127	7F	□

Fig. 7 ASCII conversion table

IV. CONCLUSIONS

The proposed algorithm possesses the potential of encrypting the data in numeric form, thus increasing the confidentiality of the information. Also the computational time required for the processing will be efficient.

REFERENCES

[1] Alfred J. Menezes, Paul C.van Oorschot and Scott A.Vanstone, *Handbook of Applied Cryptography*, 1977 by CRC Press LLC.
[2] The IEEE website. [Online]. Available: http://www.ieee.org/
[3] wikipedia.org.

