International Journal of Interdisciplinary Innovative Research &Development (IJIIRD)
ISSN: 2456-236X
Vol. 04 Issue 02 |2020

Deepkey Algorithm -*Cipher Key Generator*

[1]Gopal Kushwaha
[1]*Master student, Department of MCA, Jain Deemed-to-be University, Bengaluru, Karnataka, India*

ABSTRACT

Converting a plain text into a non-readable format to maintain confidentiality & integrity of data is called Encryption. And the technique used to decode that into a readable format is called Decryption. To encrypt & decrypt, algorithms were developed. This entire concept, entire technology is called Cryptography. Many algorithms were developed, many are cracked, and many of the algorithms are still running nowadays also. So, here I came up with the new algorithm, with a new technique, with a new concept in the algorithm.

Keyword: - Information Security, Integrity, Encryption, Decryption, Symmetric Algorithm, Cipher

1. INTRODUCTION

In Information Security, Encryption & Decryption is used to maintain data integrity & confidentiality. Cryptography is all about encryption & decryption. And cryptography comes under cryptology. In Cryptography, algorithms are there, through which security takes place. Algorithms are of 2 types: 1) Symmetric Algorithm & 2) Asymmetric algorithm. In the symmetric algorithm, public-key encryption concept is used & in Asymmetric algorithm, public key - private key algorithm concept is used. According to security researchers, the symmetric-key algorithm process fast as compare to the public key (asymmetric key) algorithm because the symmetric algorithm can't use lengthy mathematical logics & concepts. So, here I came with a new technique of key algorithm to encrypt a plain key to make secure communication. It's just converting a key to encrypted form. It's a technique which converts 16 characters to 192 characters. Yes, it's like encoding 128 bits to 1536 bits of encryption. And I'm using here 8 x 8 of the matrix. Means 64 characters is there; 26 Lowercase (small) alphabets, 26 Uppercase (CAPITAL) alphabets, 0-9 (10 numerical digits), 2 special characters. And I'm converting by doing 12 rounds of matrix & their combinations.

This is a symmetric key algorithm, where the key will be in encrypted form, and only the receiver who has this application or the one who knows the algorithm flow can decrypt it. This algorithm is a kind of Playfair algorithm, Caesars Cipher and ROT13 algorithm but not the same as that. I referred similar kind of concepts of Playfair Algorithm, Caesars Cipher and ROT13 algorithm.

2. PROPOSED SYSTEM

Proposed systems from where I got to know they can be vulnerable and some weakness which I noticed. That proposed algorithms are as follows:

1. Caesar Cipher
2. Playfair Algorithm
3. ROT13 Algorithm

Now, let's discuss these algorithms & their weakness which I noticed:

2.1 Caesar Cipher

Caesar cypher algorithm is an algorithm which performs shifting of alphabets with some certain number of positions from down or up alphabets. It's a kind of substitution cypher.

Example:

Suppose, if shifting position number is 3, then A would be replaced by D. And the position is permanent for the entire key string.

Text: HelloWorld
Shift: 3
Cypher: KHOORZRUOG

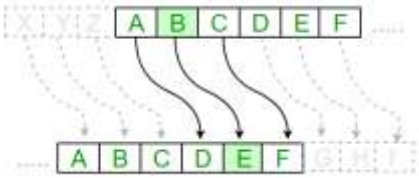

Fig 1: Overview of Caesar Cipher
Source: https://www.geeksforgeeks.org/caesar-cipher-in-cryptography

As we can see, 26 alphabets are there. So, there are 26 possibilities is there to crack the cypher key. Now, what is cypher key? "The string result which we get after encryption is called Cipher Key."

2.2 Playfair Algorithm

Playfair algorithm is also a substitution cypher. The concept of Playfair algorithm is, it swaps two alphabets position with each other in row-wise & column-wise and if both were not possible then it'll form a rectangle with 2 letters & swap the letters with horizontal corner values. Matrix is of 5x5, means 25 alphabets are there but alphabets are of 26 characters. The J will be merged with I. And the key string will be firstly distributed in the matrix then remaining alphabets will be distributed.

E.g.: Suppose key is a **monarchy,** then the matrix will be like:

M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

Fig 2: Overview of Playfair Algorithm
Source: https://www.geeksforgeeks.org/playfair-cipher-with-examples/

Plain text: instruments
Split in: 'in',' st','ru',' me','nt','sz'

Rules for Encryption:

1. If both characters are in the same column, then the next character to the present character will be swapped vertically, as follows:

M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

Fig 3: Rule 1 of Playfair Algorithm
Source: https://www.geeksforgeeks.org/playfair-cipher-with-examples/

2. If both characters are in the same row, then the next character to the present character will be swapped horizontally, as follows:

M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

Fig 4: Rule 2 of Playfair Algorithm
Source: https://www.geeksforgeeks.org/playfair-cipher-with-examples/

3. If both did not work, then the rectangle will be created, as follows:

M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

Fig 5: Rule 3 of Playfair Algorithm
Source: https://www.geeksforgeeks.org/playfair-cipher-with-examples/

2.3 ROT13 Algorithm

ROT13 means rotate 13 positions. ROT13 algorithm is the same as the Caesar cypher algorithm but shifting position is always 13, it'll never change.

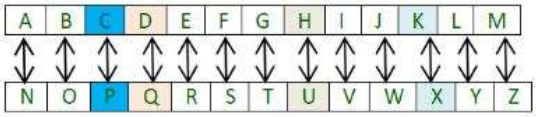

Fig 6: Overview of ROT13
Source: https://www.geeksforgeeks.org/playfair-cipher-with-examples/

If a key is a **DUCK,**

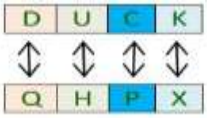

Fig 7: Example of ROT13
Source: https://www.geeksforgeeks.org/playfair-cipher-with-examples/

You're thinking what's the difference between caesar cypher & ROT13? The difference is the implementation of an algorithm. So, cracking this algorithm is possible, if analyst or cracker try all 26 possibilities to crack. So, getting this all possibilities which can crack these algorithms, I merged concepts of Caesar Cipher, Playfair Algorithm & ROT13. After that, I added some salt & spices by doing some rounds of AES Algorithm. Now you are thinking what AES algorithm is & what are rounds? AES algorithm is the asymmetric key algorithm where public key & private key system is used. An AES algorithm generates encryption by performing the same process repeatedly. It's like looping of the process till some limit. So, here also implemented this technique. Because of this, I have given name to this algorithm, which is called a Deepkey Algorithm? It generates 1536 bits of the key which is difficult to crack.

3. FLOWCHART

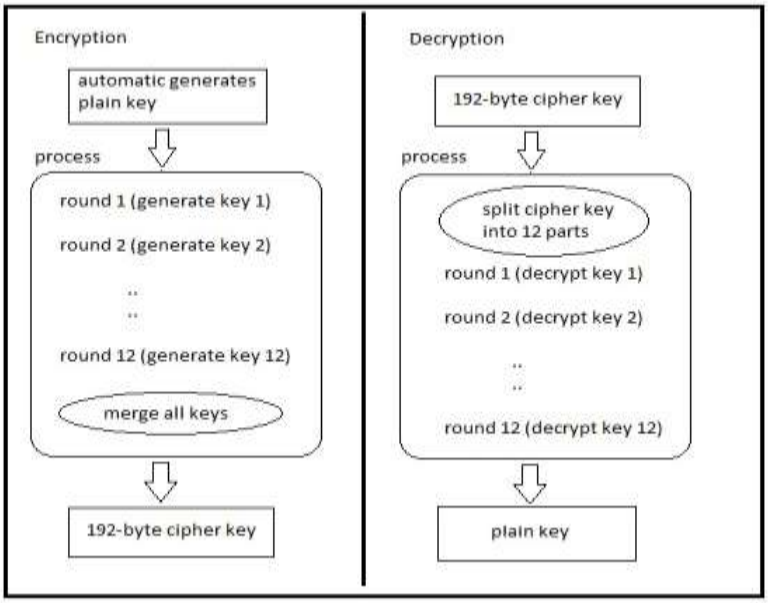

Fig 8: Flowchart of DeepKey Algorithm

4. PROBLEM WITH PROPOSED SYSTEM

I created this new algorithm on the basis of the Playfair algorithm. The problem with Playfair algorithm is that the key will easily know by any cracker, because while encrypting that key will be put the first row, then remaining blanks will be filled with remaining characters. Like if the key is "Hacker". Then it will be stored in the 5*5 matrix is like:

Table 1: Problem in Playfair algorithm

H	A	C	K	E
R	B	D	F	G
I/J	L	M	N	O
P	Q	S	T	U
V	W	X	Y	Z

But in my matrix, the key stored at different places like 1st, middle and last, column-wise & rows-wise. It's not the same as Playfair, kind of, like changing positions. Cracking Playfair cypher possibilities are 625 (25*25).

5. PROCEDURE FOR ALGORITHM (ENCRYPTION)

1) Take a string of set of all Uppercase(A-Z) Alphabets, Lowercase(a-z) alphabets, Numerical (0-9) digits,
   Special characters(@,#)
   e.g.: str = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789@#"

2) Convert string & store it into 1D-Array
3) Convert 1D-Array into 2D-array

A	B	C	D	E	F	G	H
I	J	K	L	M	N	O	P
Q	R	S	T	U	V	W	X
Y	Z	a	b	c	d	e	f
g	h	i	j	k	l	m	n
o	p	q	r	s	t	u	v
w	x	y	z	0	1	2	3
4	5	6	7	8	9	@	#

Table 2: Convert of a 1-Dimensional array to 2-Dimensional array

4) Take characters from particular positions(0,3,4,7) row-wise & column-wise, and store it into 2D-Array
   P=array of position(0,3,4,7)
   A=1D-array
   K=key

```
the loop I of A row-wise
  loop X of P
    if P == I
       loop J of A [ I ] column-wise
         loop Y of P
       if J == Y
              K = K + value at position[I][J]
```

In the end, we'll get 16 digit key. And it will be stored in 2D-Array.

5) Swap 1st row with 4th Row &
   5th Row with 8th Row
   repeat step 4

6) Swap 1st column with 4th column &
   5th column with 8th column
   repeat step 4

7) Swap 1st-row values with last row values,
   2nd row with (last – 1) row,
   3rd row with (last – 2) row,
   4th row with (last – 3) row,
   and repeat step 4

8) Swap 1st column with the last column,
   2nd column with (last – 1) column,
   3rd column with (last - 2) column,
   4th column with (last – 3) column,
   and repeat step 4

9) Swap 1st-row values with 5th-row values,
   2nd row with 6th row,
   3rd row with 7th row,
   4th row with 8th row,
   and repeat step 4

10) Swap 1st column with 5th column,
    2nd column with 6th column,
    3rd column with 7th column,
    4th column with 8th column,
    and repeat step 4

11) Now, in the selected block, increase ASCII values with 1

A	B	C	D	E	F	G	H
I	J	K	L	M	N	O	P
Q	R	S	T	U	V	W	X
Y	Z	a	b	c	d	e	f
g	h	i	j	k	l	m	n
o	p	q	r	s	t	u	v
w	x	y	z	0	1	2	3
4	5	6	7	8	9	@	#

Table 3: Select block from the array

And repeat step 4

12) Now, in a selected block, increase ASCII values with 2

A	B	C	D	E	F	G	H
I	J	K	L	M	N	O	P
Q	R	S	T	U	V	W	X
Y	Z	a	b	c	d	e	f
g	h	i	j	k	l	m	n
o	p	q	r	s	t	u	v
w	x	y	z	0	1	2	3
4	5	6	7	8	9	@	#

Table 4: Select block from the array

And repeat step 4

13) Now, in a selected block, increase ASCII values with 1

A	B	C	D	E	F	G	H
I	J	K	L	M	N	O	P
Q	R	S	T	U	V	W	X
Y	Z	a	b	c	d	e	f
G	h	i	j	k	l	m	n
O	p	q	r	s	t	u	v
W	x	y	z	0	1	2	3
4	5	6	7	8	9	@	#

Table 5: Array after increasing ASCII values

And Repeat step 4

14) Now, in a selected block, increase ASCII values with 2

A	**B**	**C**	**D**	**E**	**F**	**G**	**H**
I	**J**	**K**	**L**	**M**	**N**	**O**	**P**
Q	**R**	**S**	**T**	**U**	**V**	**W**	**X**
Y	**Z**	**a**	**b**	**c**	**d**	**e**	**f**
g	**h**	**i**	**j**	**k**	**l**	**m**	**n**
o	**p**	**q**	**r**	**s**	**t**	**u**	**v**
w	**x**	**y**	**z**	**0**	**1**	**2**	**3**
4	**5**	**6**	**7**	**8**	**9**	**@**	**#**

Table 6: Array after increasing ASCII values

And Repeat step 4

15) Again repeat step 14, increase ASCII value with +2

16) After this all, we'll get 2D-array, full of 12 keys from 12 rounds, as follows:

a. ADEHY12569aduxy#
b. Y125ADEHuxy#69ad
c. 1Y52DAHExu#y96da
d. 96daxu#yDAHE1Y52
e. ad69y#uxEHAD25Y1
f. EHAD25Y1ad69y#ux
g. ADEHY12569aduxy#
h. BEEHZ22569aduxy#
i. BEGJZ24769aduxy#
j. BEGJZ2477:advyy#
k. BEGJZ2477:cfvy{%
l. BEGJZ2477:cfvy}'

Now, we've to merge this all keys. So, we can get out encrypted keys. But, before merging of all keys, we have to make it complex. Because the last key is needed to start decryption & 1st one to get the plain key.

Now, replace 1st position with 6th position & 7th position

with 12th position. We'll get the following results.

1. EHAD25Y1ad69y#ux
2. Y125ADEHuxy#69ad
3. 1Y52DAHExu#y96da
4. 96daxu#yDAHE1Y52
5. ad69y#uxEHAD25Y1
6. ADEHY12569aduxy#
7. BEGJZ2477:cfvy}'
8. BEEHZ22569aduxy#
9. BEGJZ24769aduxy#
10. BEGJZ2477:advyy#
11. BEGJZ2477:cfvy{%
12. ADEHY12569aduxy#

Now, merge all keys, the 192-byte encrypted key will be generated, which look like:

EHAD25Y1ad69y#uxY125ADEHuxy#69ad1Y52DAHExu#y96da96daxu#yDAHE1Y52ad69y#uxEHAD25Y1ADEHY12569aduxy#BEGJZ2477:cfvy}'BEEHZ22569aduxy#BEGJZ24769aduxy#BEGJZ2477:advyy#BEGJZ2477:cfvy{%ADEHY12569aduxy#

6. PROCEDURE FOR ALGORITHM (DECRYPTION)

192-byte key needed first,

EHAD25Y1ad69y#uxY125ADEHuxy#69ad1Y52DAHExu#y96da96daxu#yDAHE1Y52ad69y#uxEHAD25Y1ADEHY12569aduxy#BEGJZ2477:cfvy}'BEEHZ22569aduxy#BEGJZ24769aduxy#BEGJZ2477:advyy#BEGJZ2477:cfvy{%ADEHY12569aduxy#

Convert this one string into 2D-array, as follows:

1. EHAD25Y1ad69y#ux
2. Y125ADEHuxy#69ad
3. 1Y52DAHExu#y96da
4. 96daxu#yDAHE1Y52
5. ad69y#uxEHAD25Y1
6. ADEHY12569aduxy#
7. BEGJZ2477:cfvy}'
8. BEEHZ22569aduxy#
9. BEGJZ24769aduxy#
10. BEGJZ2477:advyy#
11. BEGJZ2477:cfvy{%
12. ADEHY12569aduxy#

From 192-byte key, last 16-digit key is needed to start decryption.

But before it, we've to put values at their own position. Swap 1st values with 6th value & 7th value with 12th value.

It will look like,

1. ADEHY12569aduxy#
2. Y125ADEHuxy#69ad
3. 1Y52DAHExu#y96da
4. 96daxu#yDAHE1Y52
5. ad69y#uxEHAD25Y1
6. EHAD25Y1ad69y#ux
7. ADEHY12569aduxy#
8. BEEHZ22569aduxy#
9. BEGJZ24769aduxy#
10. BEGJZ2477:advyy#
11. BEGJZ2477:cfvy{%
12. BEGJZ2477:ehvy}'

Now, pick 12th position value, and proceed to decryption.

1) Convert key string into 1D-array.
2) Make a loop of 4*4

B	E	G	J
Z	2	4	7
7	:	e	H
V	y	}	'

Table 7: Array of key

3) Deduct table ASCII values with -2
Key: BEGJZ2477:cfvy{%

4) Again, deduct table ASCII values with -2
Key: BEGJZ2477:advyy#

5) Deduct table ASCII values with -1
Key: BEGJZ24769aduxy#

6) Deduct table ASCII values with -2
Key: BEEHZ22569aduxy#

7) Deduct table ASCII values with -1
ADEHY12569aduxy#

8) Swap left 2 columns with right 2 columns, like swapping 1st position table-column with 3rd position table-column & 2nd position table-column with 4th position table-column
Key: EHAD25Y1ad69y#ux

9) Swap top 2 rows with bottom 2 rows, like

swapping 1st position table-row with 3rd position table-row & 2nd position table-row with 4th position table-row
Key: ad69y#uxEHAD25Y1

10) Swap 1st column with 4th column, 2nd column with 3rd column
Key: 96daxu#yDAHE1Y52

11) Swap 1st row with 4th row, 2nd row with 3rd row
Key: 1Y52DAHExu#y96da

12) Swap 1st column with 2nd column, 3rd column with 4th column
Key: Y125ADEHuxy#69ad

13) Swap 1st row with 2nd row, 3rd row with 4th row

Key: ADEHY12569aduxy#

Now finally, we got our key 16-digit key which is ADEHY12569aduxy#

6. CONCLUSIONS

Key generation algorithm is the strongest & unbreakable. The plain key will be automatically generated every time. So, for crackers, it's difficult to crack. By using this algorithm, integrity & confidentiality should be maintained. As much as the bits are lengthy, the difficulty to crack the algorithm is difficult.

7. REFERENCES

[1] Network Security & Cryptography by Atul Kahate
[2] Websites:
   a. www.geeksforgeeks.com
   b. www.tutorialspoint.com
   c. www.javatpoint.com