PasswordManager with CipherKey

[1]Sweety Gone, [2]Raghavendra R

[1]Student, [2]Assistant Professor
Department of Master of Computer Applications
Jain (Deemed- to- be) University
Bangalore, India

***Abstract*: Transforming a plain text into non-readable format to keep a confidentially & probity of data is called Encryption. And the method used to decode that into decipherable (readable) format, is called Decoding. To encode & decode, algorithms were developed. This entire theory, entire method is called Cryptography. Many algorithms were developed, many were cracked, and many of the algorithms are still running nowadays also. So, here I came up with the new algorithm, with new method, with new idea in algorithm.**

Password managers are censorious (critical) pieces of software relied upon by users to securely store valuable and sensitive details from on-line banking passwords.

Password manager taking action like a binary (digital) safe, which securely supply user's usernames, passwords and other sensitive information. The usernames and passwords are encoded before saving them into the database using cryptography methods.

***Keywords*: Information Security, Integrity, Encryption, Decryption, Symmetric Algorithm, Cipher, Storing Password**

I.INTRODUCTION

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

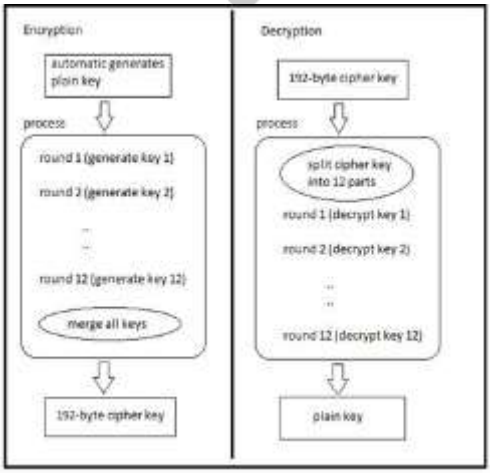
This is a symmetric key algorithm, where the key will be in encoded form, and only the receiver who has this application or the one who knows the algorithm flow can crack it. This algorithm is a kind of Playfair algorithm, Caesars Cipher and ROT13 algorithm but not as same as that. I referred interconnected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

Password manager proceed like a digital safe, which securely supply user's usernames, passwords and other sensitive details. The usernames and passwords are encoded before saving them into the database using cryptography methods.

A password manager is mainly an encipher digital vault that supply secure password login details you use to proceed towards apps and accounts on your mobile device, websites and other services.

A password manager is a program that store all your passwords, as well as other detail, in one acceptable (suitable) location with one master password. The advantages of using a password manager are: A password manager will do the task of generating the complex passwords you need to help secure your online accounts.

II. WORKING FLOW



Insert the data and perform the data pre-processing steps. Executes cryptography method to store the input data securely

III. RESULT AND DISCUSSION

- Fig 1 shows that the index page in that it asking us to input data. Input types are Application Name, Email, Mobile Number, User Name, Password. And after submitting it will take to other page
- After adding inputs and submitting one message will come that is "Record Added Successfully" as shown in Fig 2.
- Fig 3 is having 3 buttons are 1) DECRYPT 2) UPDATE 3) DELETE if we click on decrypt button it show the password in decrypted form usually no one will show the password in decrypt format. And it generates key automatically which will be hidden.

Figure 1: Input Types

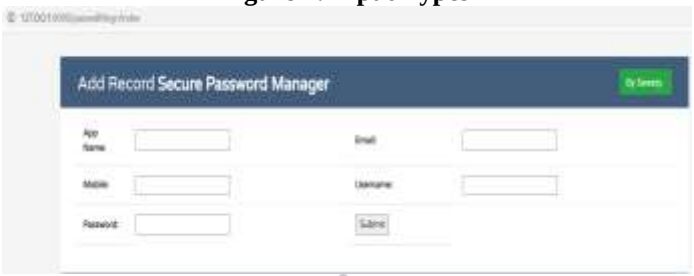


Figure 2: Adding Records And Storing Data in encrypted form

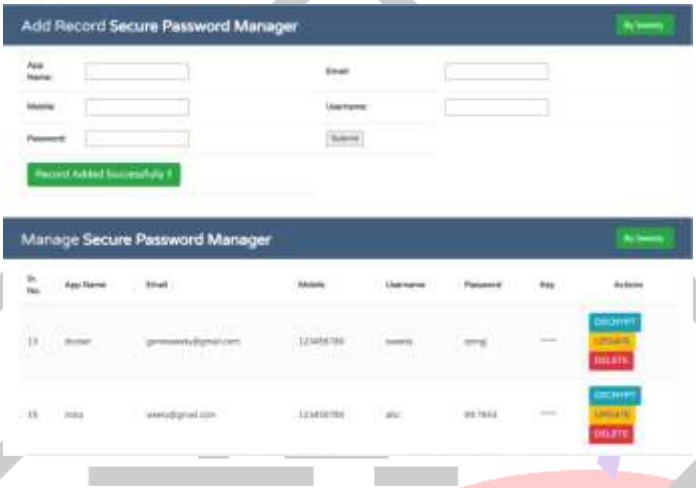


Figure 3: Decrypting Data

VI. PROBLEM WITH PROPOSED SYSTEM

I've created this new algorithm which is built on playfair algorithm. The trouble with playfair algorithm is that, key will easily know by any cracker, because while encrypting that key will be put first row, then remaining blanks will be filled with remaining characters. Like if key is "MONARCHY". Then it will be stored in 5*5 matrix is 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

J is not there in this table but it will be with I like this I/J But in my matrix, key stored at different places like 1st, middle and last, column-wise & rows-wise. It's not the same like Playfair, kind of, like changing positions. Cracking playfair cipher possibilities are

625 (25*25).

V. PROCEDURE AND ALGORITHM (Encryption)

1) Take a string of set of all Uppercase(A-Z) Alphabets, Lowercase(a-z) alphabets, Numerical (0-9) digits, Special characters (@, #) these are the string which I've taken.
e.g.: Str "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789@#"
2) Convert string & store it into 1D-Array (1 Dimensional array)
3) Convert 1D-Array into 2D-array (2 Dimensional array)
4)

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	1	2	3	4
5	6	7	8	9	0	#	@

Table1: Conversion of 1-Dimensional array to 2- Dimensional array

5) 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
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]
```

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

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

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

8) 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

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

10) 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

11) 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

12) Now, in 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	1	2	3	4
5	6	7	8	9	0	#	@

Table 2: Select block from array

And repeat step 4

13) Now, in 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 3: Select block from array

And repeat step 4

14) Now, in 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	1	2	3	4
5	6	7	8	9	0	#	@

Table 4: Iincreasing Array After ASCII values

And Repeat step 4

15) Now, in 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	1	2	3	4
5	6	7	8	9	0	#	@

Table 5: Increasing Array After ASCII values

And Repeat step 4

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

17) 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 joint this all keys. So, we can ge encoded keys. But, before uniting of all keys, we have to make it complex. Because last key is needed to start decoding & 1st one to get plain key.

Now, replace 1st position with 6th position & 7th position with 12th position. Will be getting 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, unite all keys, 192-byte encrypted key will be generated, which look like:

EHAD25Y1ad69y#uxY125ADEHuxy#69ad1Y52DAHExu#y96da96daxu#yDAHE1Y52ad69y#uxEHAD25Y1ADE
HY12569aduxy#BEGJZ2477:cfvy}"BEEHZ22569aduxy#BEGJZ24769aduxy#BEGJZ2477:advyy#BEGJZ2477:cfv
y{%ADEHY12569aduxy#

VI. PROCEDURE FOR ALGORITHM (Decryption)

192-byte key needed first,

EHAD25Y1ad69y#uxY125ADEHuxy#69ad1Y52DAHExu#y96da96daxu#yDAHE1Y52ad69y#uxEHAD25Y1ADE
HY12569aduxy#BEGJZ2477:cfvy}"BEEHZ22569aduxy#BEGJZ24769aduxy#BEGJZ2477:advyy#BEGJZ2477:cfv
y{%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	}	"

Table6: Array of key

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

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

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

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

7) Remove table ASCII values with -1 ADEHY12569aduxy#

8) Switch 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 first column with fourth column, second column with 3rd column Key: 96daxu#yDAHE1Y52

11) Swap first row with fourth row, second row with third row Key: 1Y52DAHExu#y96da

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

13) Swap first row with second row, third row with fourth row

Key: ADEHY12569aduxy#

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

VII. CONCLUSIONS

CipherKey creates algorithm in the strongest & non-breakable. Plain key will be automatic generated every time. So, for decoder it's difficult to crack. By using this algorithm, probity & clandestineness should be maintained If bits are too increased then it will be difficult to decode.

Acknowledgement

I should convey my real tendency and obligation to Dr MN Nachappa and Asst. Prof: Raghavendra R and undertaking facilitators for their effective steerage and consistent inspirations all through my assessment work. Their ideal bearing, absolute co- action and second discernment have made my work gainful.

Works Cited

[1] Burke, L. A. and H. M. Hutchins, Training transfer:Human resource development review, (2007).

[2] Gasti, Paolo and K. B. Rasmussen, "On the security of password manager database formats," *European Symposium on Research in Computer Security*, 2012.

[3] McCarney and Daniel, "Tapas: design, implementation, and usability evaluation of a password manager," 2012.

[4] Alkaldi, Nora and K. Renaud, "Encouraging password manager adoption by meeting adopter self-determination needs," *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.

[5] Stobert, Elizabeth and R. Biddle, "A password manager that doesn't remember passwords," *Proceedings of the 2014 New Security Paradigms Workshop*, 2014.

[6] Carswell and Anne, "The Canadian Occupational Performance Measure: a research and clinical literature review," *Canadian journal of occupational therapy*, 2004.

[7] Nofriansyah, Dicky and R. Rahim, "Combination Of Pixel Value Differencing Algorithm With Caesar Algorithm For Steganography," *International Journal of Research In Science & Engineering*, 2016.

[8] R. Babu and K, "An extension to traditional playfair cryptographic method," 2011.

[9] M. Sujitha and M. Pushpa, "An Encryption Algorithm for Improving Database Security using ROT & REA," *Indian Journal of Computer Science and Engineering* , 2015.