VOL. 7, NO. 1, JANUARY 2012 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

GENERALIZATION OF VIGENERE CIPHER

Yumnam Kirani Singh
Center for Development of Advanced Computing, Saltlake Sector-V, Kolkata, India
E-Mail: Yumnam.singh@cdackolkata.in

ABSTRACT

A generalized way of Vigenere cipher is proposed. Instead of using a Vigenere square for encryption and decryption, any two reversible square matrices whose rows or columns are unique are used for encryption and decryption purpose. One matrix can be easily obtained from the other and hence any one of them can be used for encryption while the other can be derived from the other for decryption. This avoids the necessity of using two separate reversible matrices for encryption and decryption. Also, a new algorithm for generation of key-stream with or without using a random symbol sequence is suggested. The key streams are generated from a small key word. The key streams are different for any slight difference of keywords either in length or content. Moreover, a key stream would be effectively random and could be made as long as we please.

Keywords: vigenere cipher, vigenere square, enciphering table, deciphering table, random key-stream, Kaisiki test.

1. INTRODUCTION

Vigenere cipher, being poly-alphabetic cipher was one of the most popular ciphers in the past because of its simplicity and resistance to the frequency analysis test of letters that can crack simple ciphers like Caesar cipher. But with the increase in the cryptanalytic skills, Vigenere cipher is no longer taken as secure cipher and is not popularly used. The most weak point of Vigenere cipher is the use of repeated words as key-streams that causes repetition of certain patterns in cipher texts at intervals equal to the length of the keyword used. Kaisiski and W. F Friedman [1, 2] suggested methods for cracking the Vigenere cipher. Several tools are also available online for cracking Vigenere cipher. Out of these tools, [2] is quite effective to find the repeated patterns and their intervals of occurrence and hence will be used in testing the strength of our proposed cipher.

Several new block ciphers such as DES, AES [3, 4, 5, 6] developed to enhance security are very complex algorithms and hence are difficult to understand and implement. Vigenere cipher is comparatively simple to understand. Encryption in Vigenere cipher is fast but the decryption process is slow. In [8], a look-up table is suggested to increase the speed during the decryption. Also, to make the Vigenere cipher more secure, a simple way for effective key-stream generation and use of random encryption and decryption table were used. In [8], the random key stream generation was based on initial random sequence and a chosen keyword.

In this paper, a better method of generating key-stream from a chosen keyword with or without using an initial random sequence is suggested. Also, a generalized method of generating a random deciphering table from any random enciphering table is described. This will enable us to store only the enciphering table for both encryption and decryption purpose. The method of obtaining deciphering table from the enciphering table is also applicable to Vigenere square.

The paper is divided into five sections. Section 2 describes about random enciphering and deciphering tables in Vigenere cipher. Also, a general way of obtaining a deciphering table from an enciphering table is given. Section 3 describes a method of generating random key stream from a chosen keyword. Better key-stream increases the tightness of security in Vigenere cipher as this makes the deciphering of the cipher text from the knowledge of the key length difficult. Experimental results are given in section 4 and conclusions in section 5.

2. GENERALIZED RANDOM ENCIPHERING AND DECIPHERING TABLES

Encryption and decryption in Vigenere cipher can be done easily using an enciphering table and its corresponding deciphering table [8]. In original Vigenere cipher only the enciphering table known as Vigenere square is used. The decryption done using the same enciphering table involves a lot of searching operations. Use of deciphering table avoids searching operation during decryption and hence makes the decryption as fast as the encryption.

The Vigenere square used in Vigenere cipher is fixed. In [1, 8], use of random table in place of Vigenere square for encryption and decryption is mentioned. But the process of generating random enciphering and deciphering tables were not generalized. Here, a generalized way of obtaining deciphering table from any enciphering table is described. From the encryption and decryption point of view, the enciphering and the deciphering tables must be reversible. That is, one must be the reciprocal or inverse of the other.

Let the enciphering table be denoted by E_t and deciphering table by D_t. These two tables are said to be reversible table if

$$E_t(P,K) = C$$

and $D_t(C,K) = P$

where P is an element of plain text and C is an element of cipher text and K is an element of key-stream.

On careful study on Vigenere cipher, we find that the reversibility of enciphering tables lies on the uniqueness of rows or columns of an enciphering table.

We do not require both the rows and the columns of an enciphering table to be unique as in Vigenere cipher. Any random or regular table whose rows or columns are unique can be used as enciphering table. Once such an enciphering table is obtained we can easily generate its corresponding deciphering table. The reversible relation between the enciphering table E_t and the deciphering table D_t can be expressed as:

$M = D_t (E_t (M, N), N)$

From the above relation, we see that N is fixed in both enciphering and deciphering table. So, the requirement in the random tables is that the columns need to be unique. That is, no two elements in a column should be the same. There is no such restriction for rows for reversible tables whose columns are unique.

So, once we have an enciphering table whose columns are unique, we can easily get the corresponding deciphering table D_t as

$D_t (E_t (M, N), N) = M$, for all M and N.

In this way, we can easily generate the deciphering table D_t from the enciphering table E_t. Table-1 shows a random enciphering table of size (26x26) and Table-2 shows its corresponding deciphering table.

Table-1. Random enciphering table E_t of size (26x26).

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

Table-2. Random deciphering table D_t of size (26x26).

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

Using these tables, enciphering and deciphering can be performed in the same way as we do in Vigenere cipher [8]. The use of the random enciphering tables makes the cipher text untraceable and hence prediction of the input from the output is almost impossibly difficult. This in addition to the use of random key stream makes the cipher undecipherable for the interceptors.

The generation of random deciphering table from random enciphering table whose rows are unique can be modified from the relation as:

$D_t(M,E_t(M,N)) = N$ for all M and N.

Also, it may be noted that once enciphering table and its corresponding deciphering table are obtained, the role of enciphering and deciphering table can be exchanged. That is, the deciphering table can be used for encryption and enciphering table for decryption.

The method of generating deciphering table is simple and fast and hence can be used for generation of deciphering table during decryption. This eliminates the requirement of putting a separate deciphering table in the cipher.

3. RANDOM KEY-STREAM GENERATION

The main weakness of the Vigenere cipher is the use of periodic key stream made by repeating a chosen keyword. The tightness of Vigenere cipher depends on the randomness of the key-stream used. Several ways to make Vigenere cipher stronger by using effective key streams were suggested in [5, 6, 7, 8]. Use of text stream of a chosen page from a mutually accepted book as a key stream is suggested in [5, 6] to increase the security of the Vigenere cipher. But there are several limitations for such key-streams to be used in practice. Use of cipher text of a repeated random sequence repeated chosen keyword as key-stream is suggested in [8]. In [7], a random sequence known as MLS of desired length is generated from a chosen keyword. The generated random sequence is used as key-stream. MLS sequence is found to be effective key-stream for Vigenere cipher.

A new random key-stream generation method, similar to the MLS generator is proposed here. Here, like MLS sequence a random sequence is generated from a chosen keyword. However, unlike MLS, the random sequence is totally changed when the length of the sequence is changed. In MLS, the longer sequence is the continuation of the shorter sequence. That is, in MLS, random sequences of different lengths generated from a given key word are not significantly different. But here for a chosen keyword, the random sequences of different lengths would be significantly different.

The algorithm is described below. In the following, the arrays are represented by combination of two letters and scalars by single letter.

Inputs: Random Symbol Sequence, Sq
Key word, Ky
Required length, L
Output: Keystream, Ks

Initialization:

```
If Sq not provided
    Sq = [1, 2, 3, …, 26];
End if
S = Integer of sum of Ky divided by 2.
N = length of the keywords
M = 1+Integer of L divided by N
K = 0
KY = 1+ (cumulative sum of Ky) mod 26
```

KeyStream Generation:

```
For I = 1 to M
    Sq = 1 + (Sq+M+N+S mod 26)
    For J = 1 to N
        d = KY (J) +S mod 26 +1
        Ks (K) = Sq (d)
        Ky(J) = Sq(d)
        S = S+ KY (J)
        K = K+1
    Next J
Next I
```

Note here that the arrays are considered unity offset arrays and hence 1 is added after mod operation. In the initialization, we test whether a random sequence of symbol is provided. If not provided, we use regular symbol sequence of 26 characters corresponding to the Roman alphabet. Initial keyword is modified so that we do not directly use the input key word for key stream generation. This will make the key stream more randomized even if the keyword like "ppppp" etc is used. Also, we modify the symbol sequence using the parameters M, N and S. This makes the symbol sequence change when we change every time we change length of the required key stream or length of the keyword N. Then we randomly compute an index d using the current keyword element and the sum of the keyword. The value symbol sequence Sq, corresponding to the computed index is output as key a stream element. The current element is updated with the output stream element. The process continues to generate a random key-stream of desired length.

4. EXPERIMENTAL RESULTS

In the experimental result, we compare the effectiveness of the key-streams of the proposed random key-generation method with the random key-streams of MLS [7]. Also, we are interested to compare the randomness of the cipher texts generated using Vigenere square and random enciphering table when a random key-stream is used. For our experiments we consider the plain text shown in Figure-1(a). The plain text is encrypted with two types of key-streams one generated using the proposed random key generation method and the other using MLS based method [8]. Two keywords "ABCD" and "WINGER" are chosen to generate four key streams K-1, K-2, K-3 and K-4. The key streams K-1 and K-2 are generated from the proposed key stream generation method from the keywords "ABCD" and the keyword "WINGER" without using any initial random sequence. The key streams K-3 and K-4 are the key streams generated from the MLS generator for the two keywords. The key-streams are shown in Figure-2 (a), (b), (c) and (d). Also, we use Vigenere square and random enciphering table (Table-1) during the encryption. Thus for a given key-stream, we get two different cipher-texts; one corresponding to the Viginere square and the other to a random enciphering Table. In this way, we get eight cipher texts (CT-1, CT-2, …, CT-8) corresponding to the four key-streams. The cipher texts are shown in Figure-3. The first four cipher texts (CT-1, …, CT-4) are obtained by encrypting the plain text using the Vigenere square and the four key-streams. The last four cipher texts (CT-5 to CT-8) are generated using random enciphering table (Table-1) and four key-streams. The cipher texts are all random in nature irrespective of whether the cipher texts are obtained using the Vigenere square or random table. That is mainly because the key-streams are all random. This shows that randomness of the cipher text is much dependent on the randomness of the key-stream used rather than the randomness of the enciphering table. However, using enciphering table will make it cracking of the cipher text more difficult.

To test the randomness in the generated cipher texts, we find any repetitive patterns in the cipher texts. We use pattern analysis tool [2] for searching any repetitive patterns in the cipher texts. The analysis results are given in Table-3.

THEFUNDAMENTALOBJECTIVEOFCRYPTOGRAPH
YISTOENABLETWOPEOPLEUSUALLYREFERREDT
OASALICEANDBOBTOCOMMUNICATEOVERANIN
SECURECHANNELINSUCHAWAYTHATANOPPONE
NTMARVINCANNOTUNDERSTANDWHATISBEING

SAIDTHISCHANNELCOULDBEATELEPHONELINEO
RACOMPUTERNETWORKFOREXAMPLE
Figure-1: Plain Text

ZNPIFHNQLDIRWEALKVTRZDTYGSOXZVGAXCF
HCOIAAUOFSZSOOILGCVZBATADIXKANDYPADZ
IDFXPOKTZWGLAYGPXCOQJVNFHUBBJMDJORY
AKHKJRJTNJQQIUDXVAWNSDTEASYKYMRGBAO
AQJOAVJRMZIZATWWDVYRUHDUVVTXYQZPEK
LAQGBHIKCJZVLTWXDJMDVAFIAIXSNPBFFZESG
KFDPCIHQJBQOAUAGUTGJYCYYHQOFH
Figure-2(a) K-1: Keyword="ABCD"

HZVOXFQVRLFGWSAKCPLKJQIKGHEOKHZUGXB
PETKSBTMTKMNPHCAGXFTOLUZWXYDCQZIZCF
SZHVDOLSVCYJPWYBWPTVOAPDNCSJAJTGHKL
DJAGOQFWEHGDEQRBPVBSDNDMBVDOYIFAUY
NOHQHNNINNMDMWDUYQMWRXFPLWWAXEDB
XCRATPMEFKMDMMQJFHRNRTITYNNNNNYGPAI
DDBOZJHPNWBEUUNLBECVXMTIDWJLCKWBKFA
Figure-2(b) K-2: Keyword= "WINGER"

SNXTMTPHVGWSWLPGJJLIJISNTGETHLGGFKKNE
SVGQDVGPIFMLQJXTEYHACBKKBGSUQMBGIUL
SKUVWXEACXVOUKARHEIESWNBWYDMKDMPV
NYLVSMLODQVJHVAPMHWCSIEPHTUEGQOISAG
XRDMDAKTDBLJLFXHFGITOKBHGXOIXWBQIVW
YXBWIGNGSRXQGPDWFPRPNOWNXBOSMNRUPX
COMXQVCJPNRYVCLCLQNPWVILHNFUOLHI
Figure-2(c) K-3: Keyword="ABCD"

RDTQFVEAHQUVTLIWEWRVTSCSIGQXVAMOVJV
YXEFWERVQLGRVHLEOEFRMUOTMBOJYNMUM
GGUCHLWJKFITAMHJKDFJTFVGGWFMWOGPSKE
VNNHRJLQQHDIXYYTKNJGHWCUKCWKIXDBWF
QDRDHJXOFPSJBVPVDULGSJMRMLUQMROEAQC
HSVQQHNRBSNJDFLNVWHXAKCVORMWEIMEDI
QBJILKRSLXJVQJNMNLLMDJLYYSHQOVWUWGW
Figure-2(d) K-4: Keyword="WINGER"

UUPXPGQKBBFCEHOQZJJCJSLQZKDBQYIGUYKA
WUKTOKZVJMMFIGEYMUMDUZUXDOORRCGCR
BELLVVLXYJFEHSBQVERAAWDZADVGSDFJBIM
WKNIXSLAVJURXXWRFQXUGUITHUTPCJCBXJOO
ZEXKYAWMRBDSONVXYKIGAYMXTIBODVSTMA
YCGCUHWLXGJDMPURHITCRQDWWALHTRAGJIF
HQHUJOLAGFZLSODNKTQUYEWQPGZTWBGX
Figure-3(a) CT-1: (Vigenere square, K-1)

MIJRXINFVTINETORHPRJZFWEZVNKFMPMLDOS
UPIBNLBHRZREPMPYRKSQJYVEONVPOGWSPZLU
HFPMAQHCCEOFQAXZJTYMFKVAIKEFCYKQCFE
SYOEMIYAUKJOUHYXTKENTOXYEMVSIOVRBZX
XMZNJIVBZORKUVEBHNUNEPRHWKWQHVZNIU
RGWDTHEWGMYVGWRUJGBNQORNGGFPPZLKD
XJEXZEEBKSVHIDNLJWKVJHJFGCNELFGE
Figure-3(b) CT-2: (Vigenere square, K-2)

BUHMIUOTRYRBEAZVAVRLZNMBMWNFIIIAMQF
UUQXNYBSUMDZHLYGHVLNXUQTQBKSZKPSQL
WJIWQYFPLYEYQINURTXVKEICRVBEVBCLBFLS
VPHJKIGQZRFEGJLTBLYAPSSLRANWNKZGXPIQN
KHJAHCFMRRCIRMPHVLDISTHGIZDXHCTJMPJRE
AXGBQBFRUYKIGNXFYPFRDSQTSCQUYHJUBHY
WYTZZYWYCGLTDJZVPXDKRSGYEEW
Figure-3(c) CT-3: (Vigenere square, K-3)

CELPPSZAFOTYHAGFFILYPDCWXWBBUTCSWRUJ
BENXKNSKQFNYPDLQKKUSAEBOKXPTRTKFLYJ
RHPWRBDULABWSEYOFJJRGORDQEFYZDUNFAV
GBVREBXZZDPPLBVEMNGFGMYCJZDQZRJZMXK
XEWYVCDZMHLSKUIOVUSAGIGXRFTWTSQUMN
XQLNRCBUZPXWNSRXOCESQPKSEPWZAMZYMV
DXDNWGDTTWGHHRGCSQNDTMNHBQCESTFI
Figure-3(d) CT-4: (Vigenere square, K-4)

VQSHHSBOQSRAOPPMQITAQJWJRHFGFPNRAUU
UPAAXPQPQAALDBYGTEVAHALAZJNKYHWGBY
SDPYQXJZLTLOUKOJVFFNMNILZJNRTHTLSVCCO
IGGYUSFTQYMMESPQUAPQBRSAXNASAWJMFPP
FMEEUCJLWGFBIEFAOQGSJKZWQADLMCCYLLB
RGCBTPNSNGHXYTDLGDCBDGEUPDIHCNOBLPC
UUOUJNYMXUTFFIQXSEXJMJXGRINWPG
Figure-3(e) CT-5: (Table-1, K-1)

KOISTBUNPNBAOIPHYSFSPKEWRNIVXKJAILFOA
AGXLWWSHUHFQENTFUGUZJSONSMXFZEPXEOV
BNTCOCXJAOZEJAFLTDELPOUNBEFTNTPPBPOJTJ
AISCUCOMFDBXUEHQNRZMPQCMFBPONMPGMK
VNIOUWIZTOEMAWIGADFXTICEUBBXCGVYISUJI
YQVSLLQSCHDTYHSBTHNCGWSNLYOHZQOGLW
OYTIOGJHIMIVGEIDYTUJFOOXPF
Figure-3(f) CT-6: (Table-1, K-2)

XQDFIYZIKTTXOOLVZFFPPBLMFKIJXJNHIPXQAC
UAQSECYJEYWQDDEYSGAQEPDKZYQDCWIECJQ
PJNGYCFUQQKEHXJNTCDDTUYOMSDBSPJEUTVI
HIUUIVNOSIQDWDANNFFJOZUUAOPYZNNQGOY
AUENPIFYBBFKHUXWFGTIUZBLLFBGGIWTHPOT
XCECYBVCXOSPGJPRHEHLTUICYEEXDTAWDQC
UGLVOCPJUMNESLLFHHPHSREGFE
Figure-3(g) CT-7: (Table-1, K-3)

AQWDHENUMFWPSOYEVSFPAUJQHKPGVXDXCY
VKGSDFTGEOMWGPQHYUTUDCMLHZKZXVHQQ
PITCIBXQYDJOWUWTBWVWTTOKOGTJEODTLUU
INZUSYFFAPGILLTTWDUOYZPARCUBSLLBCTOX
HJOGOVEQBUPIXJEOEJGJKCSSJRMJHNUXCQCLT
MIDNLDXQPTTYNETGNGWRBHKCOGEULHLQIFJ
LVGQULGDNDGYHUBCBQHVPRBPUYOIJWS
Figure-3(h) CT-8: (Table-1, K-4)

Table-3. Repetitive patterns in cipher texts.

Cipher text	Repeated words	Spacing	Key lengths	
			Actual	Suggested
CT-1	CGC	110	4	2,5,10,11
CT-2	PZL	136	6	2,4,8,17
CT-3	HVL KIG YWY	106 80 6	4	2 2,4,8,5,10,16,20 2,3,6
CT-4	YPD USA CES	28 104 43	6	2,4,7,14 2,4,8,13 NIL
CT-5	ZJN	34	4	2,17
CT-6	JAI BPO	79 31	6	NIL NIL
CT-7	XJN ECY LLF	58 140 61	4	2, 2,4,5,7,10,14,20 NIL
CT-8	JEO	57	6	3,19

From Table-3, we see that there are not many repetitions in any of the cipher texts. This shows that most of the cipher texts are highly random and hence derivation or deduction of key-length from the cipher text is not possible. By chance, the key length 4 is one of the suggested key-lengths in the case CT-3 and CT-7. But since 4 occur only once, this cannot be interpreted as the most probable key-length. However, none of the key lengths suggested is true for key-length 6. Even if a suggested key-length is assumed to be correct, the exact key word cannot be determined because key stream is not the periodic repetition of the chosen keyword. Hence, the cipher texts cannot be deciphered from the knowledge of the length of the chosen-keyword. Also, from Table-3, we see that the cipher texts (CT-1, CT-2, CT-5, CT-6) generated using the key streams of the proposed random sequence generation method have less number of repetitive patterns as compared to the cipher texts (CT-3, CT-4, CT-7 and CT-8) obtained from the key-streams of MLS. This shows the proposed method generates good random key-streams comparable to those of MLS key-streams.

5. CONCLUSIONS

A generalized model of Vigenere cipher is proposed in which any matrix whose rows or columns are unique can be used in place of Vigenere square. The matrix may be random or regular. Once such a matrix is obtained, its corresponding reversible matrix can be easily derived. Either of the two reversible matrices can be used for encryption or decryption. Use of random matrices in place of Vigenere square will increase the difficulty level of cracking the cipher. An improved random key stream generation method is also suggested to enhance the security level of the Vigenere cipher.

REFERENCES

[1] http://www.topspysecrets.com/vigenere-cipher.html

[2] http://www.simonsingh.net/The_Black_Chamber/cracking_tool

[3] David Salomon. 2003. Data Privacy and Security. Springer.

[4] Bruce Schneier. 2001. Applied Cryptography. John Wiley and Sons.

[5] Douglas R. Stinson. 1995. Cryptography, Theory and Practice. CRC Press.

[6] Michael Welchenbach. 2001. Cryptography in C and C++. A Press.

[7] Y.K. Singh, S.K. Parui. 2004. Simplet and Its Application in Signal Encryption. Multi-dimensional System and Signal Processing. 15(4): 375-394.

[8] Y. K. Singh. 2011. A Simple, fast and secure cipher. ARPN Journal of Engineering and Applied Sciences. 6(10): 61-69.