VOL. 6, NO. 10, OCTOBER 2011 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences


www.arpnjournals.com

A SIMPLE, FAST AND SECURE CIPHER

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

ABSTRACT

Vigenere cipher is a poly-alphabetic cipher once thought to be secure. The cipher is simple and easy to understand and implement. But the weak point is that the cipher uses a key stream formed by a periodic repetition of a chosen keyword. This results in the repetition of some character sequence at the multiple intervals of the length of the keyword used. By careful observation and analysis of the repeated character sequences, the key length can be deduced (as in Kaisiski Test). Once the correct length is known, the cipher text can be deciphered or decrypted. To overcome this awkwardness, a random key stream generation method is proposed. The cipher text generated using a random key is found be effective and detection of key length is almost impossible. Also, to provide more security, use of a random sequence of alphabet is also proposed for enciphering and deciphering purposes. Experimental results show that the use of random tables and random key streams makes the Vigenere cipher stronger and resistant to cipher text only attack.

Keywords: vigenere cipher, vigenere square, poly-alphabetic cipher, enciphering table, random key stream, kaisiski test.

1. INTRODUCTION

Vigenère Cipher is one of the most popular ciphers in the past. Because, it is simple to understand and implement and the cipher appeared to be unbreakable. Vigenere cipher is a poly-alphabetic cipher in which a letter or symbol in a plain text is represented by two or more different letters or symbols in the resulting cipher text. This makes the Vigenere cipher resistant to the frequency analysis test of letters that can crack simple ciphers like Caesar cipher. Also, because of the lack of cryptanalytic skills or enthusiasms in the past, quite long time, it is considered as undecipherable cipher.

Vigenere ciphere indeed has two weak points. One is the used of Vigenere square and the other is the use of a repeated keyword as key stream. The second point is more vulnerable to cipher text attack using careful analysis of repetition of similar patterns in a cipher text. Using this weak point, Kaisiski, could crack the Vigenere cipher [1, 2]. Mention may be made that W. F Friedman could also crack the Vigenere cipher using statistical information of the probability of occurrence of letters in cipher texts, a method known as Index of Coincidence. There are also various tools available online about cracking Vigenere cipher. Out of this tool, [2] is used found to be quite effective and used in testing our proposed ciphers.

Several new modern ciphers have been developed after the cracking of the Vigenere cipher. Among these ciphers, DES, AES are the popular ones [3, 4, 5, 6] as these have been accepted as encryption standard by NIST. But they are complex algorithms to understand and implement. Moreover, the complexities of DES and AES limit them for use in the image and video encryption algorithms. We are interested to develop a simple and computationally fast encryption method without involving complex mathematics. Our aim is to improve the Vigenere cipher to make it a fast and secure cipher without sacrificing much of its simplicity. Computationally, in Vigenere cipher, enciphering is much faster compared to deciphering. This is because deciphering requires a lot searching operations. To make the deciphering as fast as enciphering, we suggest the use of a deciphering table to avoid searching. Moreover, to improve the security of Vigenere cipher, we propse a better key stream generation method. The key streams are generated from a chosen key word and a random sequence of symbols. The generated key streams have no similarity with the chosen keyword. It is found that the using better key stream the security of the Vigenere cipher is very much improved.

The paper is divided into five sections. Section 2 describes about the enciphering and deciphering table in Vigenere cipher. Also, generation and use of enciphering and deciphering table from a random sequence of symbols is described. Section 3 describes a method of generation of the longer key stream from a chosen shorter keyword. This makes the tracing of the original key length difficult. Moreover, 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. REGULAR AND RANDOM SEQUENCE ENCIPHERING AND DECIPHERING TABLES

Vigenere cipher is a poly-alphabetic cipher and was thought to be undecipherable quite a long time. It is simple and easy to implement. It is based on a fixed table known as Vigenere square for enciphering and deciphering. This table can be formed by cyclic rotation of the sequence of ordered or regular alphabet sequence, “ABCDE…XYZ”. The table shows parallel patterns along the diagonal lines. All elements except the first and the last elements along the diagonal lines are the same.

The Vigenere square is shown in Table-1. The encryption using the Vigenere square is fast but the decryption is slow. The reason for slowness during decryption is that during decryption we have to search the elements of the column and then find the corresponding row, which contains the cipher element. The searching operation causes the decryption computationally slow.

We can avoid the searching operation in order to make the decryption process as fast as the encryption

process. For this, we require to generate the deciphering table. The deciphering table is generated from the column of the regular alphabet sequence "ABCDE…XYZ". The second column is obtained from the first column by bottom-to-top rotation by one element. Similarly any subsequent columns can be obtained from its preceding column by bottom-to-top rotation. The deciphering table is shown in Table-2. It can be seen that the elements along the off-diagonal lines in the table are the same.

Let the enciphering table be denoted by E_t and deciphering table by D_t. These two tables are reversible tables. In other words, if $E_t(x, y) = z$, then $D_t(z,y) = x$, where x, y, z are symbols in the two tables.

That is, if z is the symbol at the intersection of the xth row and yth column of the enciphering table, then the element at the intersection of zth row and the yth column in the deciphering table D_t is x. For example, from Table-1 and Table-2, we have

$E_t(C, B) = D$, $D_t(D, B) = C$

If C is an element of a plain text and B is element of key stream, then D is the cipher text obtained using the E_t.

Table-1. Regular enciphering Table (26x26).

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

Table-2. Regular deciphering Table (26x26).

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

Similarly, we can get back the plain text C from the deciphering table D_t directly using the cipher text element D and the key element B. Thus, we avoid searching during decryption. Note here that, we can use D_t for encryption and E_t for decryption. However, the cipher text generated by the tables shows regular patterns, which provide some clues to cryptanalysts to crack the cipher.

To make the cryptanalysts confused, we can use enciphering and deciphering tables generated from a random sequence of symbols or alphabets. This will make the cracking of the resulting cipher text more difficult. The generation of the enciphering and deciphering table can be similarly done from any random sequence of alphabets.

Let us consider the following random sequence of 26 alphabets.

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

The enciphering and the deciphering tables are shown in Tables 3 and 4, respectively.

Table-3. Random enciphering Table (26x26).

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

Table-4. Random deciphering Table (26x26).

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

Once we generate the random enciphering and deciphering tables, encryption and decryption can be done in the similar way [1]. For Example, E_t (K, I) denotes the element at the intersection of Kth row and Ith column in Et (Table-3), that is B. Similarly, D_t (F, I) = K, the element at the intersection of the Fth row and Ith column in the decryption table (Table-4). It is found that the cipher text generated using the enciphering or deciphering table generated from a random sequence of alphabet is more difficult to crack.

3. EFFECTIVE KEY STREAM GENERATION

The main weakness of the Vigenere cipher is the use of periodic key stream made by repeating a chosen keyword. This causes sequences of letters repeat in the cipher text at the multiple intervals of length of keyword used. This provides clue for deducing the length of the keyword. Once the key length is known the cipher text can be decrypted [2]. In order to make Vigenere cipher stronger, a new key stream generation method known as autokey was also purposed. But the key-stream generated by the autokey is not random but mere concatenation of the plain text to a chosen keyword and can be cracked from the bi-gram, tri-gram pattern analysis.

A way for generation of random permutation of random alphabet is given in [6]. There suggests two ways of generating a random sequence of length 676 from a chosen random sequence of alphabets. But it did not discuss about the way of generating key stream from shorter chosen keyword. In [5, 6], using of text stream of a chosen page from a mutually accepted book as a key stream is suggested to increase the security of the Vigenere cipher. It has several limitations for practical implementations and the security cannot be guaranteed

because the possible key streams are known. We want a simpler way of generating effective key streams from a chosen keyword. The key stream should not be related to the key stream to make deduction of key word and key length very difficult or impossible.

To generate a better key stream, we consider any chosen random sequence of symbols or letters. The random sequence is made equal to the length of the plain text to be encrypted by repeating itself. Next, an initial key stream (of length equal to the length of the plain text) is formed from a chosen keyword by repeating itself. The initial key stream and the random stream are added under modulo M operation, where M is 26 in our case, as we consider here only 26 symbols. This gives the final key stream, which can be used for encryption of the plain text using enciphering table. Similarly for decryption, we first generate the final key stream and then decrypt the cipher text using the deciphering table.

Note here that lengths of chosen the random sequence of alphabet and the chosen keyword must not be equal. In that case, the final key stream would be a periodic repetition. Usually, random key sequence must be much longer than the keyword chosen. As we use modulo M operation to get the final key stream from the initial random stream and initial key streams which are formed by repetition of a chosen random sequence and a chosen keyword, the final key stream is periodic at certain interval which depend on the lengths of the chosen random sequence and chosen keyword. If random sequence is of length M, and the chosen keyword is of length K, then the key stream is repeated after L, where L is the largest common multiple of M and K. This way, the periodicity of the key stream is increased by multiple of the initial random sequence used. Moreover, the key stream is quite different from the initial keyword chosen.

Algorithm for key stream generation:

Inputs: Length of the plain or cipher text (L)
: Keyword, Ky
: Initial random sequence, Rs
Output: Key stream equal to the length of the plain text or cipher text, Ks

Steps:

a) Concatenate Rs repeatedly to make it equal to the length of plain text.
b) Concatenate Ky repeatedly to make it equal to the length of the plain text.
c) Generate the final key stream, Ks from the initial random streams and the initial key streams using the modulus of length of Rs.

When the key repeats in longer sequence, the repetition in the cipher text may still be longer. This will increase the difficulty level of deciphering the cipher text. It is difficult to crack the cipher text generated using key streams with periodic repetition longer than 26. The key streams generated by the proposed key stream generator will have periods much larger than 26 and hence it would be difficult to crack the cipher text.

In another way, we can generate the key stream using the random number generator known as MLS generator [7] This sequence takes a positive array, i.e., an array whose all elements are all positive numbers. The ASCII values of the characters of any chosen keyword can be considered as positive array for generating a long random key stream without repetition. Such a key stream can be considered as practical one-time pad key stream. Such a key stream would make the Vigenere cipher an undecipherable cipher.

4. EXPERIMENTAL RESULTS

The original Vigenere cipher based on the key-stream formed by repetition of a chosen key word is not secure. The cipher text generated has patterns of combination of certain characters repeated at multiple intervals of the length of the chosen keyword. By analysis of the intervals of the repeated patterns, the key length can be determined. Once the length is determined, cracking of the cipher text can be done by careful analysis of the frequency of the occurrence of the characters in the cipher text. A good analysis tool for determination of key length from the cipher text of Vigenere cipher is given in [2]. This tool lists the repeated patterns and intervals of repetition in a given cipher text. Also, it suggests probable key lengths by factoring the interval of repetition of patterns. The tool is effective to determine the shorter key lengths usually less 15. We use the tool for the analysis of the repeated patterns in the cipher text generated for different key-streams and methods suggested. It is found that in most cases the tool fails to determine the correct key-length of the chosen key word. This is because the repetition in the key-stream is different from the length of keyword. Even if the length of the keyword is known, the cipher text cannot be cracked using the conventional methods. So, the main purpose of using the cracking tool is just to find the repetition of patterns in the cipher text generated by the proposed methods.

The plain text we consider for our experimentation is given below. Two keywords “ABCD” and “WINGER” are chosen. Four key streams K-1, K-2, K-3 and K-4 are used to generate eight cipher texts (CT-1, CT-2… CT-8). The key streams K-1 and K-3 are generated from the proposed key stream generation method from the keywords “ABCD” and “WINGER”. The key streams K-2 and K-4 are the key streams from the MLS generator for the two keywords. It is found that the key streams K-1and K-3 are repetitive with respective intervals of 52 and 78 as expected. Cipher text generated with such key stream with large repetition interval is very difficult to analyze and crack. We can increase the interval of repetition in the generated key stream if we use longer keyword and longer initial random sequence whose lengths are relatively primes. Another way to get key stream with no repetition is to use the MLS generator. We see that no part is repeated in the key streams (K-2 and K-4) generated by MLS generator. So, the key stream K-2

and K-4 can be considered as random sequence with no repetition. The cipher texts generated using the key streams K-2 and K-4 can be considered undecipherable, as they are similar to practical one-time pad cipher.

To generate cipher texts, we consider the only the encryption tables (Table-1 and Table-3). The cipher text CT-1 is generated using the Table-1 and key stream K-1. The cipher text CT-2 is generated using the Table-3 and K-1. The table and key stream used in generating a cipher text is written inside a bracket for notational purpose. E.g., (Table-3, K-4) means the cipher text is generated using the Table-3 and key stream K-4. As two tables and four key streams are considered, we get eight different cipher texts. The cipher texts CT-1 to CT-4 are generated from the keyword "ABCD" and the cipher text CT-5 to CT-8 are generated from the keyword "WINGER". The cipher texts are analyzed for any repetitive patterns and their intervals. The analysis result of the cipher texts is shown in Table-5.

Plain text:
THEFUNDAMENTALOBJECTIVEOFCRYPTOGRAPH
YISTOENABLETWOPEOPLEUSUALLYREFERREDT
OASALICEANDBOBTOCOMMUNICATEOVERANIN
SECURECHANNELINSUCHAWAYTHATANOPPONE
NTMARVINCANNOTUNDERSTANDWHATISBEING
SAIDTHISCHANNELCOULDBEATELEPHONELINEO
RACOMPUTERNETWORKFOREXAMPLE

Keyword="abcd"

Key-stream: k-1
IRAFFBUJTIXXWSORNERHJLNEZYKTYDHDSHVK
VVYUMPPGPFLNLCBAIRAFFBUJTIXXWSORNERH
JLNEZYKTYDHDSHVKVVYUMPPGPFLNLCBAIRAF
FBUJTIXXWSORNERHJLNEZYKTYDHDSHVKVVY
UMPPGPFLNLCBAIRAFFBUJTIXXWSORNERHJLNE
ZYKTYDHDSHVKVVYUMPPGPFLNLCBAIRAFFBUJ
TIXXWSORNERHJLNEZYKTYDHDS

MLS key-stream: k-2
SNXTMTPHVGWSWLPGJJLIJISNTGETHLGGFKKNE
SVGQDVGPIFMLQJXTEYHACBKKBGSUQMBGIUL
SKUVWXEACXVOUKARHEIESWNBWYDMKDMPV
NYLVSMLODQVJHVAPMHWCSIEPHTUEGQOISAG
XRDMDAKTDBLJLFXHFGITOKBHGXOIXWBQIVW
YXBWIGNGSRXQGPDWFPRPNOWNXBOSMNRUPX
COMXQVCJPNRYVCLCLQNPWVILHNFUOLHI

Keyword= "winger"

Key-stream: k-3
EYLIJROOGNZLSZZURULMWQPSVFVWCTBIFMX
YRCJXQFJLCKNBHJMDMHUKSGWXPPIAAIIWAJT
VFSYHDOEYLIJROOGNZLSZZURULMWQPSVFVW
CTBIFMXYRCJXQFJLCKNBHJMDMHUKSGWXPPIA
AIIWAJTVFSYHDOEYLIJROOGNZLSZZURULMWQ
PSVFVWCTBIFMXYRCJXQFJLCKNBHJMDMHUKS
GWXPPIAAIIWAJTVFSYHDOEYLIJRO

MLS key-stream: k-4
RDTQFVEAHQUVTLIWEWRVTSCSIGQXVAMOVJV
YXEFWERVQLGRVHLEOEFRMUOTMBOJYNMUM
GGUCHLWJKFITAMHJKDFJTFVGGWFMWOGPSKE
VNNHRJLQQHDIXYYTKNJGHWCUKCWKIXDBWF
QDRDHJXOFPSJBVPVDULGSJMRMLUQMROEAQC
HSVQQHNRBSNJDFLNVWHXAKCVORMWEIMEDI
QBJILKRSLXJVQJNMNLLMDJLYYSHQOVWUWGW

Keyword: ="abcd"

Cipher text: ct-1 (Table-1, k-1)
BYEKZOXJFMKQWDCSWITARGRSEABRNWVJJHK
RTDQNATCGQQPGHQQEWGLJZTOJETVOAXSIEIU
AXLFEKGMXYQKEGIOYXJKGGCXIPYPBGGSAVZ
NXJDOAXKEXJFSCVRJBLSNAZWDAYWHQGWKYI
ZLNYPGBXSNNYPPTCEDJWTNJGLTEWLWJOIZUP
DNMCRRBQFODFUZVXJSFPQTGIJWRAJPNMCISJP
LJVWJMQLSIAIKDXCXJNPOQYPWOW

Cipher text: ct-2 (Table-3, k-1)
UKYOKIZXOVTNEKVEVMQMOBARUUVSYGTHTF
VQKSFYLENOBHWJNMBYNRPNKNTXVYOBDXRG
HMNMEPGYRTEKIELTQAFILAJNNNIDABWSBSED
MAWXNFTMKRIKPGRUMATXNOWDNVWMIGFEQ
HVIQXDYTARPIRTWZXPCPHMNSNYXJJCIEENTIM
AUWKWVWSQUFOCMGUXXLALSVBEOTNFALBP
WVUQRNPCXPNUXHERGLMIREUCNKPONIVHAH

Cipher text: ct-3 (Table-1, k-2)
LUBYGGSHHKJLWWDHSNNBRDWBYIVRWEUMW
KZUCANZEHIGQTJFHEYBHTJLUUVKVMEJYVQSX
MXEGKMVHFGECKYPILTFJSUQMJVDWRHAFHDP
IVLDZUGCSFXVWUZLXZZQEZIAPFMBEZQBWHP
UKVQFPABOLONJYSLAZTLXFCUHTAKPXPJIJZEL
DTWQJGNAJZXGCQAQRFJYRXRXUSDQCYICBNW
ZBEMCLDZGSOGCPPJJDNFNZYRCUAASM

Cipher text: ct-4 (Table-3, k-2)
EUGBNJDFGUPEEFOLFCTUOSHSBDLSHVBMSSKU
BLWAWIQOBYNONWJGDEEZRYWSXMRTFYQEJV
ZVQSVTNHSYGTCPJZCXBRXQVPMFESILNIWAQM
DKXYNUROETVUXPIYJHAJQDANLXYAJSNSABTL
EOVDVFJITXZRYMKKJKXMHFKSUWKXKPKXBDE
NECHJVLTWEONEDHCXUZFQAKHRKQNKJNGTN
YGWYGNOYRLFSUXWRPOFNEGKAQRLLOV

Keyword: "winger"

Cipher text: ct-5 (Table-1, k-3)
XFPNDEROSRMESKNVAYNFELTGAHMURMPOW
MMFPKBQEJWLDVRUDXBHAWFOMYQXAAGREN
MNRNWOTSQHOWGCLVMSCPZBBZELTHZWLFAE
KWMFIEPLFKZDBAYCWKUQRYUEPIHFMBFOUDS
TKMEDVENBUWREBIHSLURHYLOMAJHOTQVSSS
HMSYTZCIPAYYCEUVIISZBJTQDITGNLVOYFWQA
QQSCXWUNXRDUPUBMNNNMRTJIMRFIVLUYCS

Cipher text: ct-6 (Table-3, k-3)
KNWEDHFHFAFVURKWMFTOBUEQMVYMCLPZS
BXNPRANWNVPFXAHREMILHSOVIHKLLTZYEVF
ZCGFCUFFANSBPMKEVPASJGZEGOAIPODWVHYL
QBXEJRKWGUKGVTTHOZYBXAFTBQOCRWUKUX
POMYWUXEZRUQVUDUFDFDAVMTDHKELOUQD
VEFYYSPALCBDBYPAQRYGEQMDJXDCPPOZJHY
LEQOPCHBFKCOXARUVFWCLJCTHGFXMOPXJUR

Cipher text: ct-7 (Table-1, k-4)
WZLCUTZRTKRIIHCVLYQERJXZDTEUIQTBUUDZ
OCSGUWNZKIKMPOSCQNPTLZIMBBZJUCRQJEW
HGIHWQPEMZSDRJWODVXJAKWHGZESUHBEPIR
MCZMSTISIQNEDFIHLAPCWWUGEUDRIYMDVAS
UKDLGJVDITIALEYBANRRDYRYGZKIBEURYAGD
OTTTVARLJCHZGIWWNUAYMEZUTVZNHQVHZQ
BBLQMLALGQDRQKEQQGPORECOGZLPJF

Cipher text: ct-8 (Table-3, k-4)
ILKYKQIDPTOFCFNQCDQFUWSQEDDOVCLBYXV
NOVXXRLQJMCLFRGERRFUQTQYBMGGPAAFWJ
UZPTPOXXERKDYLUITBEPCGFQPEHEDUOWOLTL
MUTCTADZOWKDDKXMVIXKBRGGMWWKGVPC
QDXEZVNLLRWOILAQEJVQGKQGZVRBVFTYCGY
AHSBNJWATEUORVOWQPZJGIIBFEQEKVDMDW
WICYYCLQUKNAKJOOAUZQGTGPHXTDRZENWCD

Table-5. Repetitive patterns in cipher texts.

Cipher text	Repeated words	Spacing	Suggested key lengths
CT-1	XJF	109	NIL
	OAX	47	NIL
	NYP	8	2,4,8
	NMC	29	NIL
CT-2	HMN	89	NIL
CT-3	LDZ	110	2,5,10,11
CT-4	ZRY	97	NIL
CT-5	ELT	70	2,5,7,10,14
CT-6	RKW	96	3,5,9,15
	EQM	169	13
	HYL	103	NIL
CT-7	ZKI	123	3
	HWG	23	NIL
CT-8	NIL	NIL	NIL

From Table-5, we see that CT-1 has only 4 patterns of three characters (XIF, OAX, NYF, NMC), which repeat at different intervals (109, 47, 8, 29), none of which have common factors. By chance, the possible factors of 8, 2, 4, 8 are suggested as possible key lengths. This is not true for all the cipher texts generated using the key streams of the proposed algorithm. The cipher texts CT-5 and CT-6 are generated the key streams generated using the proposed method. Only one pattern is repeated in CT-5 and three patterns repeated in CT-6. However, none of the suggested key lengths is true. The key streams have repetition intervals of 52 and 78 but none of the repeated patterns repeats at the multiple of 58 or 78. Hence from the interval of the repeated patterns, none is the key length that can be used for cracking the cipher texts.

The cipher texts CT-3, CT-4, CT-7 and CT-8 are obtained from the key streams of MLS generator. CT-2 and CT-4 have only one pattern repeated and no key length is suggested. CT-6 has three patterns repeated at different intervals, none of which have common factor to suggest a correct key length. None of the suggested key lengths in CT-6 is true. There is no repetition of pattern in CT-8. The key streams generated from the MLS generator have no repetition. Hence no correct key length can be obtained from the analysis of the repeated patterns of such cipher texts.

From the occurrence of the patterns in the cipher texts considered, we see that very few patterns are randomly repeated from which the determination of key length is not possible. Also, it can be seen that repetition of patterns is less in case cipher texts (CT-2, CT-4, CT-6 and CT-8) generated from Table-3. CT-2 and CT-4 has only one pattern each at the intervals 89 and 97. CT-6 has only three patterns where as CT-8 has no pattern.

Even if the suggested key length is assumed to be correct keyword length as in CT-1 (4 corresponding to the keyword “ABCD”), the cipher text cannot be deciphered because key stream is not the periodic repetition of the chosen keyword.

5. CONCLUSIONS

A simple way of improving Vigenere cipher is suggested to make it fast and secure. Deciphering of Vigenere cipher is made fast by introducing deciphering tables to avoid the search operations. The security of the Vigenere cipher is improved by using a better key stream generation method and an enciphering table generated from a random sequence. The key stream generator can generate a key stream of longer period from a chosen keyword. The generated key stream improves the security of the Vigenere cipher to a great extent. It is found that the security of the Vigenere cipher depends mostly on the use of the effectiveness of key stream. The more random sequence is the key stream the more is the difficulty to crack the generated cipher text. Use of the random key stream generated from MLS would make the Vigenere cipher an undecipherable 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++. Apress.

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