Enhancing Security of Vignere Cipher using Modified RC4

Ashish Shah
Department of Computer
Science & Engineering
Manipal Institute of Technology,
Manipal University

ABSTRACT

Encryption is a way to enhance the security of a message or file by scrambling the contents so that it can be read only by someone who has the right encryption key to unscramble it. Encryption does not of itself prevent interception, but denies the message content to the interceptor. RC4 is one of the most popular encryption algorithms and finds its applications in many security protocols such as Wi-Fi Protocol Access (WPA) and Wired Equivalence Privacy (WEP). Although being a versatile encryption technique, RC4 faces numerous loopholes and weaknesses. Encrypted messages can sometimes be broken by cryptanalysis, also called codebreaking. It uses three secret keys- two secret keys K1 and K2 as seeds for Enhanced RC4 and K3 as the key for Vigenère Cipher substitution. It also uses two S-Boxes S1 and S2. Both of them contain N elements from 0 to N-1. Although being a versatile encryption algorithm, Vigenère Cipher is not resistant to the Kasiski Attack. Through Cryptanalysis, one can determine the frequency of each letter and find a pattern in the Cipher text as well as the length of the key. In this paper, we propose an enhancement to the RC4 algorithm by converting it into a product cipher. Then we attempt encryption combining the 2 encryption techniques, Modified RC4 and Vigenere in a systematic manner.

General Terms

Cipher, Security, Stream Ciphers, Product Ciphers

Keywords

Keywords are your own designated keywords which can be used for easy location of the manuscript using any search engines.

1. INTRODUCTION

Stream ciphers process one bit or one byte at a time for encryption or decryption. One of the cornerstones of a stream cipher is the pseudorandom bit generator. The pseudorandom bit generator takes a key as the input and produces a stream of random bits as the output using a deterministic algorithm. This stream of random bits is known as keystream. The keystream is then combined, one bit or one byte at a time with the plaintext to produce the corresponding ciphertext. RC4 is a prime example of stream cipher which is widely used in many security protocols such as Wi-Fi Protocol Access (WPA) and Wired Equivalence Privacy (WEP). These protocols use RC4 because it is fast, utilizes less resource and is easy to implement.

The RC4 algorithm which was initially proposed in 1987 uses a variable length key and its operations are byte oriented. It uses a deterministic algorithm to produce a random permutation. The RC4 algorithm can be divided into two phases: Key Scheduling Algorithm (KSA) and Pseudo Random Generation Algorithm (PRGA). KSA makes use of the variable length key to initialize a 256 Bytes array S. This operation is known as the initialization of the S-block. The key is then used to produce a random permutation of the initialized array S. This marks the end of the KSA phase.

Once the array S has been initialized, the key is no longer used.

PRGA phase now begins. It produces a random sequence of words from the permutation in S known as the key stream. During the decryption process, the key stream is then XORed with the plaintext to produce the ciphertext. During decryption, the ciphertext is XORed with the keystream to produce the plaintext.

1.1 Key Scheduling Algorithm

```
for i = 0 to 255 S [i] =
i; j = 0;

for i = 0 to 255

j = ( j + S [i] + K [i % key_length] ) %
256; swap (S[i], S[j]);
```

1.2 Pseudo- Random Generation Algorithm

```
i = 0, j = 0;

while ( true )

i = ( i + 1 ) % 256;

j = ( j + S[ i ] ) % 256;

swap ( S[i], S[j] );

t = ( S[i] + S[j] ) % 256; k= S[t];
```

For encryption, the keystream k is XORed with the next byte of plaintext to produce the ciphertext. In case of decryption, the keystream is XORed with the ciphertext to produce the plaintext. However, this algorithm suffers from many weaknesses that have been exposed by various cryptanalysis attacks. The cryptanalysis of RC4 can be broadly divided into two categories: attacks focused on exploiting the randomness of KSA and attacks focused on exploiting the properties of the internal states of PRGA. Fluhrer discovered a major weakness in the RC4 algorithm i.e. it is possible to completely attack RC4 if some portion of the secret key is known.

Vigenère Cipher occurs from some flaws. The most important of them is the Kasiski examination which is also called the Kasiski test. It takes advantage of the fact that repeated words may be encrypted using the same key letters, leading to repeated groups in the cipher text.

2. RELATED WORK

Vigenère Cipher uses polyalphabetic cipher which was considered to be unbreakable and secure till 1917. But later it was broken by Kasiski and Friedman. They exploited the repeating nature of the key to break it. If a cryptanalyst knows the length of the key, then Vigenère Cipher can be treated as multiple Caesar Ciphers which can be easily broken. The Kasiski test can help determine the key length. Over the years, many improvements have been suggested to improve the security of Vigenère Cipher. Authors in used Linear Feedback Shift Registers for improving security of Vigenère Cipher for encryption and decryption using Vigenère Cipher. However, this enhancement still does not provide resistance to Kasiski attack. In this paper, we try to enhance the security of Vigenère Cipher by converting it into a product cipher, thereby providing resistance to Kasiski attack.

3. PROPOSED TECHNIQUE

3.1 Modified RC4 Algorithm

The proposed algorithm is an extension of the Improved RC4 Stream Cipher Algorithm. It uses three secret keys- two secret keys K1 and K2 as seeds for Enhanced RC4 and K3 as the key for Vigenère Cipher substitution. It also uses two S-Boxes S1 and S2. Both of them contain N elements from 0 to N-1. The Key Scheduling Algorithm is the same as original RC4 except that it uses two S-boxes instead of one, as proposed in the Enhanced RC4 Algorithm.

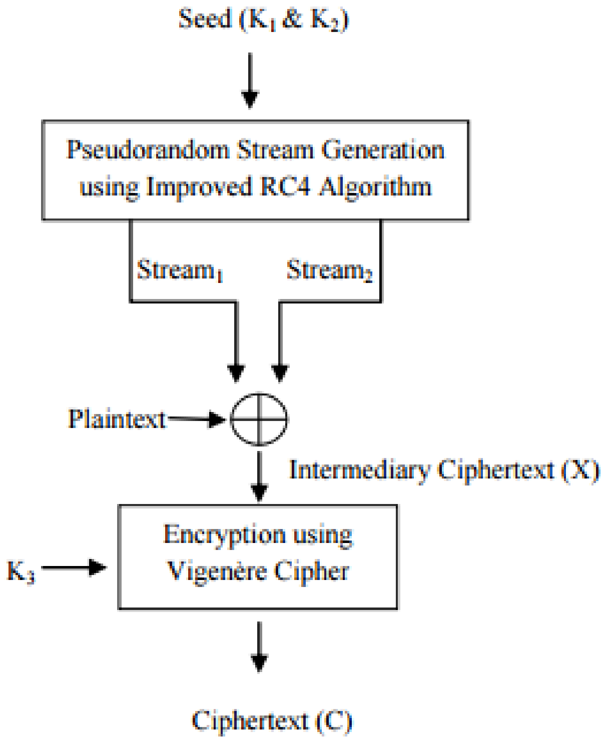

Seed (K_1 & K_2)
Pseudorandom Stream Generation using Improved RC4 Algorithm
Stream$_1$
Stream$_2$
Plaintext
Intermediary Ciphertext (X)
K_3
Encryption using Vigenère Cipher
Ciphertext (C)

Fig 3.1 Encryption Process using Proposed Algorithm

In PRGA two output streams are obtained from S1 and S2. The output streams are XORed with each other. The resulting stream is then XORed with the plaintext P, to obtain the intermediary ciphertext, X. This intermediary ciphertext is then fed as the input of Vigenère Cipher. In this final phase of the encryption process, substitution on the intermediary ciphertext X takes place using the key K3. This gives us the final ciphertext

C. The encryption process is stated below.

In the algorithm proposed, Vigenère Cipher is used in the final phase of the encryption process to perform substitution.

Vigenère Cipher is a polyalphabetic stream cipher.[8] Each character of the intermediary ciphertext X is encrypted using K3 as the key. This final phase of encrypting using Vigenère Cipher can be summarized as follows-

Encryption: Ca = (Xa + ka) mod 256

Where C = C0…..Cn is the Ciphertext, X = X0…..Xn is the Intermediary Ciphertext and K3 = k0…..km is the key used.

Decryption process is similar to encryption obeying the laws of symmetric cryptography algorithms. The ciphertext C is first decrypted using Vigenère Cipher with K3 as the key. The output of this process is the intermediary plaintext Y. In the next phase, keys K1 and K2 are used as the seed for the Pseudorandom Stream Generator using Improved RC4. Two output streams are obtained as the output of the stream generation phase.

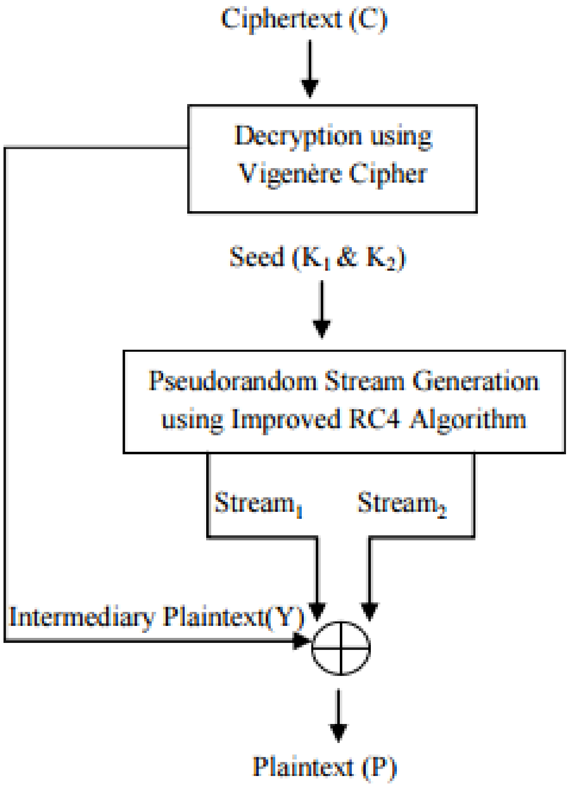

Ciphertext (C)
Decryption using Vigenère Cipher
Seed (K_1 & K_2)
Pseudorandom Stream Generation using Improved RC4 Algorithm
Stream$_1$
Stream$_2$
Intermediary Plaintext(Y)
Plaintext (P)

Fig 3.2 Decryption Process using Proposed Algorithm

The output streams are XORed with each other. The resulting stream is then XORed with the intermediary plaintext Y to give the final plaintext P.

The proposed algorithm can be summarized as follows- Encryption:

```
for i=0 to 255 S1[i]=i;
  S2[i]=i;
  j1 = j2 = 0;
  for i = 0 to 255
```

```
j1 = ( j1 + S1[i] + K1[i] ) mod 256;
swap ( S1[i], S1[ j1]);
j2 = ( j2 + S2[i] + K2[i]) mod 256;
swap(S2 [i], S2[j2]); i = jl= j2=a=0; while ( true
) i = ( i+1 )%256;
jl = jl + S1[i];
swap ( S1[i], S1[ j1] ); j2= j2 + S2[i];
swap ( S2 [i], S2[ j2] )
Stream1= S1 [ (S1 [i]+ S1[j1]) mod 256 ]; Stream2= S2 [
(S2 [i]+ S2[ j2]) mod 256 ]; swap ( S1 [S2 [j1]], S1 [S2 [j2]]
);
swap ( S2 [S1 [j1]], S2 [S1 [j2]] );
X[a] = Stream1 XOR Stream2 XOR P[a];
C[a] = (X[a] + K3 [a]) mod 256
a=a+1;
```

Decryption:

```
for a=0 to length(Y) – 1 if K3 [a] < C[a]
then Y[a] = (C[a] – K3[a]) mod 256
else
Y[a] = (256 + C[a] – K3[a]) mod 256
for i=0 to 255
S1[i]=i; S2[i]=i;
j1 = j2 = 0;
for i = 0 to 255
j1 = ( j1 + S1[i] + K1[i] ) mod 256;
swap (S1[i], S1[ j1]);
j2 = (j2 + S2[i] + K2[i]) mod 256;
swap (S2 [i], S2[j2]); i = jl= j2 =
a= 0; while ( true )
i = ( i+1 )%256;
jl = jl + S1[i];
swap ( S1[i], S1[ j1] ); j2= j2 + S2[i];
swap ( S2 [i], S2[ j2] )
Stream1= S1 [ (S1 [i]+ S1[j1]) mod 256 ]; Stream2= S2 [ (S2
[i]+ S2[ j2]) mod 256 ]; swap( S1 [S2 [j1]], S1 [S2 [j2]] );
swap( S2 [S1 [j1]], S2 [S1 [j2]] );
P[a] = Stream1 XOR Stream2 XOR Y[a]; a=a+1;
```

4. VIGNERE CIPHER

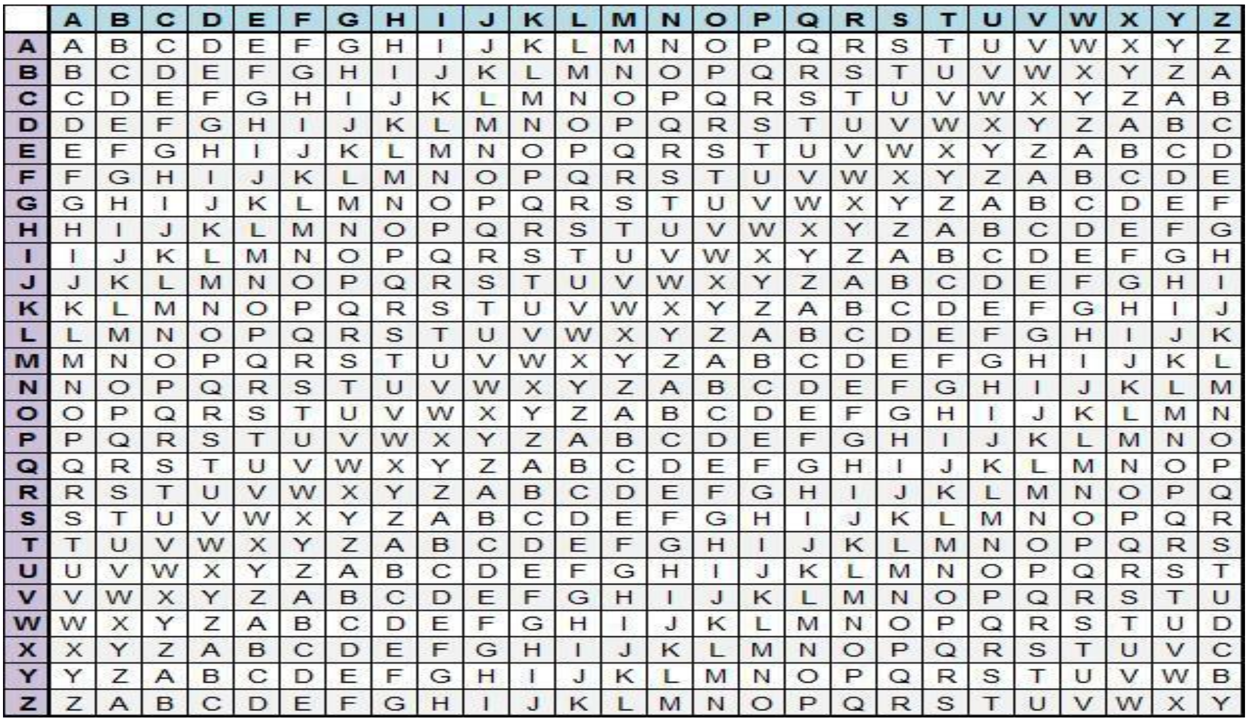
Vigenère Cipher is a polyalphabetic cipher. No permutation is involved in the encryption process. This exposes Vigenère Cipher to Kasiski attack. The attack is based on finding repetitive patterns in the cipher text.[9] The repetitive pattern should be three characters or more in length. If such is the case, then we can conclude that the distance between repetitive patterns is likely to be the key length or integral multiples of the key length.[10].

5. EXPERIMENTAL RESULT

5.1 Example

In Example – 1, let the plaintext and keys for encryption be as follows.

Plaintext	WEA**RED**ISCOVE**RED**SAVEYOURSELF
Key	DECEPTIVEDECEPTIVEDECEPTIVE



	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	T	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	T	U	D
X	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	C
Y	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	B
Z	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

Using the Vigenère Cipher encryption square on plaintext using K1, we get the 1st intermediary cipher text as follows –

C’ = ZIC**VTW**QNGRZG**VTW**AVZHCQYGLMGJ

It can be seen as the intermediary cipher text has the pattern

‘VTW’ repeating twice. This makes it susceptible to Kasiski attack. To make this resistant to Kasiski attack, we proceed with the next phase of encryption. [10] Cryptography and State-of-the-art Techniques Mohiuddin Ahmed,T. M. Shahriar Sazzad, Md. Elias Mollahmaterial

Plaintext	WEA**RED**ISCOVE**RED**SAVEYOURSELF
Key 1	BALL
Key 2	BAT
Key 3	DECEPTIVE
Ciphertext	zzaXK.y´¢H_WD”rpXX?\Al]A

6. CONCLUSION

The algorithm proposed in the paper enhances the security of Improved RC4 algorithm by imposing substitution, thereby converting it into a product cipher. Time taken for encryption/decryption using the proposed algorithm is marginally more than the Improved RC4 Algorithm. Experimental results show that there is a mere 0.8% - 1% increase in the time required for encryption/ decryption. These characteristics of the proposed algorithm make it a better candidate for practical applications as compared to the Improved RC4 Algorithm. In the future we can expand this idea so that this encryption scheme is used to encrypt not only text but also images. With more levels of encryption, through more substitution and transposition we can make this scheme much more secure.

7. REFERENCES

[1] Atul Kahate (2009), Cryptography and Network Security,2nd Edition, McGraw-Hill.

[2] OverviewofCryptographyhttp://www.garykessler.net/library/crypto.html.

[3] William Stallings: “Cryptography and Network Security: Principles and Practices” 4th Edition,

[4] Martin, Keith M. (2012). Everyday Cryptography. Oxford University Press. p. 142. ISBN 978-0-19-162588-6.

[5] CryptanalysisofVigenèreCipherhttp://www.nku.edu/~christensen/section%2012%20vigenere%20cryptanalysis.pdf

[6] Abdul Razaq ,Yasir Mahmod, Faroq Ahmed, Ali Hur : Strong Key Machanism Generated by LFSR based

Vigenère Cipher – 13th International Arab Conference on Information Technology (December 2012)

[7] Yumnam Kirani Singh :Generalization of Vigenère CipherARPN Journal of Engineering and Applied Sciences (January 2012)

[8] Alpha-Qwerty Cipher: An Extended Vigenère Cipher-Advanced Computing, An International Journal, Vol. 3, No. 3, May 2012.

[9] Cryptography Based E-Commerce Security: A Review Shazia Yasin,Khalid Haseeb,Rashid Jalal Qureshi