ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 1, Issue 10, December 2012

A cryptosystem based on Vigenère cipher with varying key

Quist-Aphetsi Kester

***Abstract*— Privacy is one of the key issues information Security addresses. Through encryption one can prevent a third party from understanding raw data during signal transmission. The encryption methods for enhancing the security of digital contents has gained high significance in the current era of breach of security and misuse of the confidential information intercepted and misused by the unauthorized parties.**

This paper sets out to contribute to the general body of knowledge in the area of cryptography application and by developing a new way of implementing Vigenère cipher encryption algorithm by automatically changing the cipher key after each encryption step. The new method will use successive keys that will be dependent on the initial key value during the encryption process. The algorithm ultimately makes it possible for encryption and decryption of the text. The ciphertext will have different encryption key pattern and the Vigenère cryptosystem will be more difficult to decipher using frequency attack. The implementation will be done using java programming.

***Index Terms*— Cryptography, Encryption, Vigenère, key.**

I. INTRODUCTION

The broaden utilization of digital media for information transmission through secure and unsecured channels exposes messages sent via networks to intruders or third parties. Encryption of messages in this modern age of technology becomes necessary for ensuring that data sent via communications channels become protected and made difficult for deciphering. Enormous number of transfer of data and information takes place through internet, which is considered to be most efficient though it's definitely a public access medium. Therefore to counterpart this weakness, many researchers have come up with efficient algorithms to encrypt this information from plain text into ciphers [1].

In cryptography, encryption is the process of transforming information using an algorithm to make it unreadable to anyone except those possessing special knowledge, usually referred to as a key. The result of the process is encrypted information. The reverse process is referred to as decryption [2]. There two main algorithmic approaches to encryption, these are symmetric and asymmetric. Symmetric-key algorithms [3] are a class of algorithms for cryptography that use the same cryptographic keys for both encryption of plaintext and decryption of cipher text. The keys may be identical or there may be a simple transformation to go between the two keys. The keys, in practice, represent a shared secret between two or more parties that can be used to maintain a private information link [4]. This requirement that both parties have access to the secret key is one of the main drawbacks of symmetric key encryption, in comparison to public-key encryption. Typical examples symmetric algorithms are Advanced Encryption Standard (AES), Blowfish, Tripple Data Encryption Standard (3DES) and Serpent [5].

Asymmetric or Public key encryption on the other hand is an encryption method where a message encrypted with a recipient's public key cannot be decrypted by anyone except a possessor of the matching private key, presumably, this will be the owner of that key and the person associated with the public key used. This is used for confidentiality. [6]. Typical examples of asymmetric encryption algorithms are Rivest Shamir Adleman (RSA),Diffie–Hellman key exchange protocol and Digital Signature Standard(DSS), which incorporates the Digital Signature Algorithm (DSA)

Modern day cryptography entails complex and advance mathematical algorithm are applied to encryption of text and cryptographic techniques for image encryption based on the RGB pixel displacement where pixel of images are shuffled to obtained a cipher image [7].

This research is aimed at contributing to the general body of knowledge in the area of the application of cryptography by developing a new way of implementing Vigenère cipher encryption algorithm by automatically changing the cipher key after each encryption step. The successive keys will be dependent on the initial key value during the encryption process. The algorithm ultimately makes it possible for encryption and decryption of the text and also makes the Vigenère cryptosystem more difficult against frequency attack using varying keys. The paper has the following structure: section II consist of related works, section III of the methodology, section IV The algorithm section V Implementation, section VI Results and Analysis and section VII concluded the paper.

II. RELATED WORKS

Caesar cipher, also known as the shift cipher, is one of the simplest and most widely known classical encryption techniques. It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet. For example, with a shift of 3, A would be replaced by D, B would become E, and so on. The encryption step performed by a Caesar cipher is often incorporated as part of more complex schemes, such as the Vigenère cipher, and still has modern application in the

***Quist-Aphetsi Kester**, Digital Forensics Department, Faculty of Informatics, Ghana Technology University College. Accra, Ghana, +233 209822141*

ROT13 system. As with all single alphabet substitution ciphers, the Caesar cipher is easily broken and in modern practice offers essentially no communication security.[10]

The encryption can also be represented using modular arithmetic by first transforming the letters into numbers, according to the scheme, A = 0, B = 1... Z = 25. [11] Encryption of a letter by a shift n can be described mathematically as, [12]

$En(x) = (x + n) \bmod 26$

Decryption is performed similarly,

$Dn(x) = (x - n) \bmod 26$

The Vigenère cipher is a method of encrypting alphabetic text by using a series of different Caesar ciphers based on the letters of a keyword. It is a simple form of polyalphabetic substitution [8][9]. The Cipher spoils the statistics of a simple Caesar cipher by using multiple Caesar ciphers. The technique is named for its inventor, Blaise de Vigenère from the court of Henry III of France in the sixteenth century, and was considered unbreakable for some 300 years [13].

Vigenère can also be viewed algebraically. If the letters A–Z are taken to be the numbers 0–25, and addition is performed modulo 26, then Vigenère encryption E using the key K can be written,

$Ci = EK(Mi) = (Mi + Ki) \bmod \{26\}$

and decryption D using the key K,

$Mi = DK(Ci) = (Ci - Ki) \bmod \{26\}$,

whereas M = M0 … Mn is the message, C = C0 …ts Cn is the ciphertext and K = K0 … Km is the used key.

Thus Given m, a positive integer, P = C = (Z26)n, and K = (k1, k2… km) a key, we define:

Encryption:

$ek(p1, p2\ldots pm) = (p1+k1, p2+k2\ldots pm+km) \pmod{26}$

Decryption:

$dk(c1, c2\ldots cm) = (c1-k1,\ c2-k2\ \ldots\ cm-km) \pmod{26}$

Example:

Plaintext: C R Y P T O G R A P H Y

Key: L U C K L U C K L U C K

Ciphertext: N L A Z E I I B L J J Ishould accompany your final submission.

	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	V
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	W
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	X
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

Fig 1 The Vigenère square

A modified form of the Vigenère cipher, the alpha-qwerty cipher extended the original 26 character Vigenère cipher to a 92 characters case sensitive cipher including digits and some other symbols commonly used in the English language and can be written from a computer keyboard. The alpha-qwerty cipher also changes the mapping sequence used in the Vigenère cipher. The mapping takes from a extended alphabet sequence to extended qwerty keyboard sequence. To decrypt the code reverse mapping takes place (compliment of encryption) that is from extended QWERTY keyboard to extended alphabet sequence. In short this proposed version extends and rearranges the original Vigenère table, therefore making it much more complex than the existing one. The greater character set allows more type of messages to be encrypted like passwords. It also increases the key domain and hence provides more security [15].

The algebraic description of the extended version is similar to that of the original cipher. It uses modulo 92 instead of modulo and cipher text Ci is derived using a sequence different from plain text sequence Pi.

$Ci = EK(Pi) = (Pi + Ki) \bmod 92$

and decryption D,

$Pi = DK(Ci) = (Ci - Ki) \bmod 92$

where, P = P0…Pn is the message,

C = C0….Cn is the ciphertext and K = K0.....Km is the used key.

Friedrich Kasiski was the first to publish a successful general attack on the Vigenère cipher. Earlier attacks relied on knowledge of the plaintext, or use of a recognizable word as a key. Kasiski's method had no such dependencies. He published an account of the attack, but it's clear that there were others who were aware of it. Babbage was goaded into breaking the Vigenère cipher when John Hall Brock Thwaites submitted a "new" cipher to the Journal of the Society of the Arts. Thwaites challenged Babbage to break his cipher encoded twice, with keys of different length. Babbage succeeded in decrypting a sample, "The Vision of Sin", by Alfred Tennyson, encrypted according to the keyword "Emily", the first name of Tennyson's wife. Studies of Babbage's notes reveal that he had used the method later published by Kasiski [11] [16].

III. Methodology

The new method employs the Vigenère square and key in its encryption process but the successive keys used will be dependent on the initial key value during the encryption process. That is the key varies as it is used in the encryption process. The first step key will be different from the second step key but the second step key will be as a result of a function that operated on the first step and so forth.

The algorithm ultimately makes it possible for encryption and decryption of the text and also makes the Vigenère cryptosystem more difficult against frequency attack. This is as a result of varying keys employed for each encryption process. A software program was written to demonstrate the effectiveness of the algorithm using java programming language and cryptanalysis performed on the ciphertext.

IV. THE MATHEMATICAL ALGORITHM

If the letters A–Z are taken to be the numbers 0–25, and addition is performed modulo 26, then Vigenère encryption E using the key K can be written,

Ci = EK (Mi) = (Mi + Kni) mod {26}

and decryption D using the key K,

Mi = DK(Ci) = (Ci - Kni) mod {26},

Thus Given m, a positive integer, P = C = (Z26)n, and K = (k1, k2… km) a key, we define:

Encryption:

ek(p1, p2… pm) = (p1+kn0, p2+kn1…pm+kmn) (mod 26)

where kni+1=f(kni , si) mod 26

i=0,1,2…n

s=x : [a, b]= {x ∈ I: a ≤ x ≥ b ,a=0, and b=25}

Decryption:

dk(c1, c2… cm) = (c 1-kn0, c 2-kn1… c m-kmn) (mod 26)

V. IMPLEMENTATION

The algorithm was implemented using java programming. A table of Vigenère cipher was generated and a chosen key was used to encrypt the plaintext. The key varies after each successive encryption in order to eliminate the repetition of the same key and plaintext encryption results within the cipher text.

With the encryption process, a table of Vigenère cipher was created and the plaintext is accepted from the user inter interface as an input as well as the key. The key is then used to operate on the message which is the plaintext to produce a ciphertext.

The first encryption step will yield a length ciphertext , Lci=Lki

The second to the nth encryption step will yield a length ciphertext , Lci=Lki-1 + Lki

Where i=1, 2, 3…n and Lk = length of key.

The will vary by a shift during each encryption process.

Below is the java codes written to implement the algorithm. The decryption process follows the similar trend but with an inverse operation in order to decipher the cipher text.

A. *Encryption*

```
int tableRowSize = 26;
 int tableColumnSize = 26;

 int vignereTable[][] = new int[26][26];
  Scanner input = new Scanner(System.in);
  for (int rows = 0; rows < tableRowSize; rows++){
    for (int columns = 0; columns < tableColumnSize; columns++){
        vignereTable[rows][columns] = (rows + columns) % 26;
    }
  }
  System.out.println("Enter the plaintext");
 String plainText = input.nextLine();
  plainText = plainText.toUpperCase();
  System.out.print("Enter the key: ");
 String key = input.nextLine();
 key = key.toUpperCase();
 String cipherText = "";
 int keyIndex = 0;
 for (int ptextIndex = 0; ptextIndex < plainText.length();
 ptextIndex++){
  char pChar = plainText.charAt(ptextIndex);
  int asciiVal = (int) pChar;
  if (pChar == ' '){
     cipherText += pChar;
     continue;
  }
  if (asciiVal < 65 || asciiVal > 90){
    cipherText += pChar;
    continue;
  }
  int basicPlainTextValue = ((int) pChar) - 65;
  char kChar = key.charAt(keyIndex);
  kChar+=1;
  int basicKeyValue = ((int) kChar ) - 65;
  int tableEntry = vignereTable[basicPlainTextValue][basicKeyValue];
  char cChar = (char) (tableEntry + 65);
  cipherText += cChar;
  keyIndex++;
  if (keyIndex == key.length())
    keyIndex = 0;
 }
  System.out.println(" cipher text is "+cipherText);
```

B. *Decryption*

```
int tableRowSize = 26;
 int tableColumnSize = 26;
 int vignereTable[][] = new int[26][26];
 Scanner input = new Scanner(System.in);
 for (int rows = 0; rows < tableRowSize; rows++){
   for (int columns = 0; columns < tableColumnSize; columns++){
      vignereTable[rows][columns] = (rows + columns) % 26;
    }
  }
  System.out.println("Enter the cipher text");
  String cipherText = input.nextLine();
  cipherText = cipherText.toUpperCase();
  System.out.print("Enter the key: ");
  String key = input.nextLine();
  key = key.toUpperCase();

  String plainText = "";
  int keyIndex = 0;
  for (int ctextIndex = 0; ctextIndex < cipherText.length();
 tIndex++){
    char cChar = cipherText.charAt(ctextIndex);
    int asciiVal = (int) cChar;
    if (cChar == ' '){
       plainText += cChar;
       continue;
```

```
        }
        if (asciiVal < 65 || asciiVal > 90){
            plainText += cChar;
            continue;
        }
        int basiccipherTextValue = ((int) cChar) - 65;
        char kChar = key.charAt(keyIndex);
         kChar+=1;
        int basicKeyValue = ((int) kChar ) - 65;
        for (int pIndex = 0; pIndex < tableColumnSize; pIndex++){
            if (vignereTable[basicKeyValue][pIndex] == basiccipherTextValue){
                char potcChar = (char)
    nereTable[basicKeyValue][pIndex] + 65);
                char potpChar = (char) (pIndex + 65);
                plainText += potpChar;
            }
        }
        keyIndex++;
        if (keyIndex == key.length())
            keyIndex = 0;
             }
         System.out.println(" plain text is "+plainText);
```

VI. Results and Analysis

The program written was used to encrypt a message and the result was analyzed with the existing Vigenère cipher. The Plaintext was encrypted by the software and analyzed below by applying the Kasiski attack.

The plain text = "IN THE FOREST THERE ARE MANY TREES WITH THE SAME HEIGHT. FOR EXAMPLE MANY"

The keyword was ="TREE"

A. *Results from existing Vigenère Cipher*

Key: TREE

Message: IN THE FOREST THERE ARE MANY TREES WITH THE SAME HEIGHT. FOR EXAMPLE MANY

Ciphertext: BE XAV JHIIWM XLXII TII FRRC KVIXJ ABKL MYI LRQI YIMZYX. WSV VBEFGPI DERR

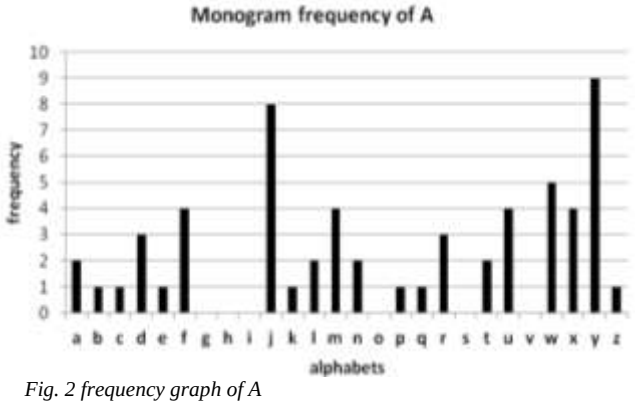


Fig. 2 frequency graph of A

B. *Result from proposed algorithm*

Key: TREE

Message: IN THE FOREST THERE ARE MANY TREES WITH THE SAME HEIGHT. FOR EXAMPLE MANY

Ciphertext: CF YMY XTWYKY YBWWJ UJJ RUFD YLWJX QAYM NZJ XUEJ MYALMN. XTW YPFRJDJ RUFD

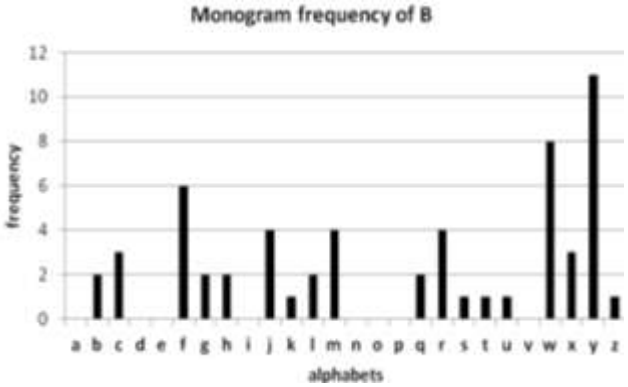
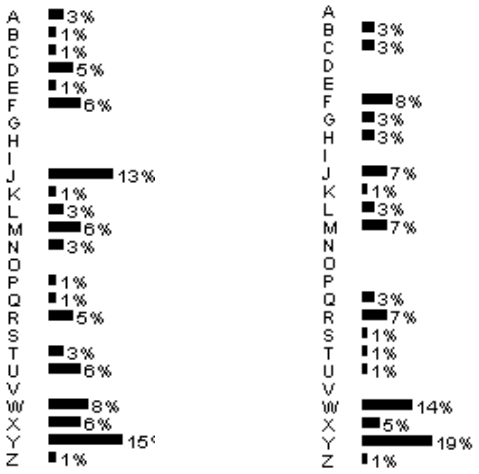


Fig 3 frequency graph of B



(a) A (b) B

Fig 4 Percentage of alphabet frequency in A (a) and B (b) respectively

The fig 2 and fig 3 represent the monogram frequency graph of A which is the result of the ciphertext from existing Vigenère cipher and B which is the result of the ciphertext from the new proposed method of encryption using the modified Vigenère cipher.

Below are the statistical and cryptanalysis results performed on the two ciphertexts.

C. *Index of Coincidence of A and B*

The I.C. of a piece of text does not change if the text is enciphered with a substitution cipher. It is defined as:

$$I.C. = \frac{\sum_{i=A}^{i=Z} f_i(f_i - 1)}{N(N-1)}$$

Where fi is the count of letter i (where i = A, B... Z) in the ciphertext, and N is the total number of letters in the ciphertext.

Table 1: Cryptanalysis of A and B

Cryptanalysis	A	B
Incidence Coincidence	0.0590	0.0631
Keyword Length	1	1
Chi-squared statistic against English distribution	415.9592	1008.8847
Chi-squared statistic against uniform distribution	56.0170	62.1864
Statistical data: Variance	14.0450	15.5918
Statistical data: Standard deviation	3.7477	3.9487
Present alphabet Entropy	4.1361	3.9564

From the analysis the index of coincidence (IC) of A and B was calculated to be to be 0.0590 and 0.0631respectively. For a normal English text of alphabet of A-Z, the Variance is normally 14.50603 and the standard deviation of 3.80868. The results indicated there is a larger deviation in B than in A.

The table below shows the N-Gram analysis of A which is the result of the ciphertext from existing Vigenère cipher and B which is the result of the ciphertext from the new proposed method of encryption using the modified Vigenère cipher. The N-Gram analysis indicates the frequency of monogram, bigram as well as trigram with their respective percentages of occurrence within the ciphertexts.

Table 2: N-Gram Analysis of A

Nr.	Histogram			Bigram			Trigram		
1.	I	11	18.64%	II	3	5.08%	RQI	1	1.69%
2.	R	5	8.47%	YI	2	3.39%	RCK	1	1.69%
3.	X	5	8.47%	BE	2	3.39%	RRC	1	1.69%
4.	V	4	6.78%	RR	2	3.39%	SVV	1	1.69%
5.	M	3	5.08%	SV	1	1.69%	VBE	1	1.69%
6.	Y	3	5.08%	TI	1	1.69%	TII	1	1.69%
7.	L	3	5.08%	VB	1	1.69%	QIY	1	1.69%
8.	B	3	5.08%	RQ	1	1.69%	PID	1	1.69%
9.	E	3	5.08%	PI	1	1.69%	LXI	1	1.69%
10.	W	2	3.39%	MX	1	1.69%	LRQ	1	1.69%
11.	F	2	3.39%	MY	1	1.69%	MXL	1	1.69%
12.	A	2	3.39%	MZ	1	1.69%	MYI	1	1.69%
13.	J	2	3.39%	QI	1	1.69%	MZY	1	1.69%
14.	K	2	3.39%	RC	1	1.69%	VIX	1	1.69%
15.	Z	1	1.69%	VJ	1	1.69%	VJH	1	1.69%
16.	C	1	1.69%	XL	1	1.69%	YIL	1	1.69%
17.	D	1	1.69%	XJ	1	1.69%	XWS	1	1.69%
18.	Q	1	1.69%	XW	1	1.69%	YIM	1	1.69%
19.	G	1	1.69%	YX	1	1.69%	YXW	1	1.69%
20.	P	1	1.69%	ZY	1	1.69%	ZYX	1	1.69%
21.	H	1	1.69%	XI	1	1.69%	XLX	1	1.69%
22.	S	1	1.69%	XA	1	1.69%	XJA	1	1.69%
23.	T	1	1.69%	LX	1	1.69%	WMX	1	1.69%
24.				VV	1	1.69%	VVB	1	1.69%
25.				WM	1	1.69%	WSV	1	1.69%
26.				WS	1	1.69%	XAV	1	1.69%
27.				VI	1	1.69%	XII	1	1.69%
28.				LM	1	1.69%	LMY	1	1.69%
29.				FG	1	1.69%	KVI	1	1.69%
30.				EX	1	1.69%	EXA	1	1.69%

Table 3: N-Gram Analysis of B

Nr.	Histogram			Bigram			Trigram		
1.	Y	9	15.25%	WY	2	3.39%	UFD	2	3.39%
2.	J	8	13.56%	JX	2	3.39%	TWY	2	3.39%
3.	W	5	8.47%	JR	2	3.39%	RUF	2	3.39%
4.	U	4	6.78%	WJ	2	3.39%	JRU	2	3.39%
5.	X	4	6.78%	MN	2	3.39%	XTW	2	3.39%
6.	M	4	6.78%	TW	2	3.39%	UJJ	1	1.69%
7.	F	4	6.78%	UF	2	3.39%	WJU	1	1.69%
8.	R	3	5.08%	MY	2	3.39%	WJX	1	1.69%
9.	D	3	5.08%	RU	2	3.39%	WWJ	1	1.69%
10.	T	2	3.39%	XT	2	3.39%	YXT	1	1.69%
11.	N	2	3.39%	FD	2	3.39%	UEJ	1	1.69%
12.	A	2	3.39%	YM	2	3.39%	RJD	1	1.69%
13.	L	2	3.39%	YP	1	1.69%	ZJX	1	1.69%
14.	C	1	1.69%	YX	1	1.69%	YYB	1	1.69%
15.	Z	1	1.69%	YY	1	1.69%	WYK	1	1.69%
16.	B	1	1.69%	ZJ	1	1.69%	WYP	1	1.69%
17.	Q	1	1.69%	UE	1	1.69%	YBW	1	1.69%
18.	K	1	1.69%	UJ	1	1.69%	YKY	1	1.69%
19.	P	1	1.69%	WW	1	1.69%	YLW	1	1.69%
20.	E	1	1.69%	YB	1	1.69%	YMN	1	1.69%
21.				XU	1	1.69%	YAL	1	1.69%
22.				YK	1	1.69%	XUE	1	1.69%
23.				XQ	1	1.69%	XQA	1	1.69%
24.				YL	1	1.69%	YPF	1	1.69%
25.				YA	1	1.69%	QAY	1	1.69%
26.				NZ	1	1.69%	YMY	1	1.69%
27.				EJ	1	1.69%	NZJ	1	1.69%
28.				FR	1	1.69%	FRJ	1	1.69%
29.				FY	1	1.69%	FDY	1	1.69%
30.				JD	1	1.69%	FYM	1	1.69%

VII. CONCLUSION

From the analysis the index of coincidence (IC) between a string and that same string with its first few characters deleted (sometimes called a shift of the string) was obtained for A to be 0.0590 and B to be 0.0631 which indicates a stronger approach of the new algorithm. For a normal English text of alphabet of A-Z, the Variance is normally 14.50603 and the standard deviation of 3.80868. For stronger implementation of the new algorithm, the length of the key have to be long and the variation function of the key must be very difficult to be predicted.

REFERENCES

[1] Kester, Quist- Aphetsi., & Danquah, Paul. (2012). A novel cryptographic key technique. In Adaptive Science & Technology (ICAST), 2012 IEEE 4th International Conference on (pp. 70-73).

[2] Abraham Sinkov, Elementary Cryptanalysis: A Mathematical Approach, Mathematical Association of America, 1966. ISBN 0-88385-622-0

[3] Nicolas Courtois, Josef Pieprzyk, "Cryptanalysis of Block Ciphers with Overdefined Systems of Equations". pp267–287, ASIACRYPT 2002

[4] Delfs, Hans & Knebl, Helmut (2007). "Symmetric-key encryption". Introduction to cryptography: principles and applications. Springer, 2007

[5] Mullen, Gary & Mummert, Carl. Finite fields and applications. American Mathematical Society. p. 112. 2007

[6] IEEE 1363: Standard Specifications for Public-Key Cryptography

[7] Kester, Q. A., & Koumadi, K. M. (2012, October). Cryptographie technique for image encryption based on the RGB pixel displacement. In Adaptive Science & Technology (ICAST), 2012 IEEE 4th International Conference on (pp. 74-77). IEEE.

[8] Bruen, Aiden A. & Forcinito, Mario A. (2011). Cryptography, Information Theory, and Error-Correction: A Handbook for the 21st

Century. John Wiley & Sons. p. 21. ISBN 978-1-118-03138-4. http://books.google.com/books?id=fd2LtVgFzoMC&pg=PA21.

[9] Martin, Keith M. (2012). Everyday Cryptography. Oxford University Press. p. 142. ISBN 978-0-19-162588-6. http://books.google.com/books?id=1NHli2uzt_EC&pg=PT142.

[10] Encryption. Wellesley college Computer Science Department lecture note retrieved from : http://cs110.wellesley.edu/lectures/L18-encryption/

[11] Caesar cipher. Retrieved from http://en.wikipedia.org/wiki/Caesar_cipher

[12] Luciano, Dennis; Gordon Prichett (January 1987). "Cryptology: From Caesar Ciphers to Public-Key Cryptosystems". The College Mathematics Journal 18 (1): 2–17. doi:10.2307/2686311. JSTOR 2686311.

[13] Wobst, Reinhard (2001). Cryptology Unlocked. Wiley. pp. 19. ISBN 978-0-470-06064-3.

[14] Vigenère cipher. Retrieved from http://en.wikipedia.org/wiki/Vigenère_cipher

[15] Rahmani, M. K. I., Wadhwa, N., & Malhotra, V. (2012). Advanced Computing: An International Journal (ACIJ). Alpha-Qwerty Cipher: An Extended Vigenere Cipher, 3 (3), 107-118.

[16] Franksen, O. I. (1985) Mr. Babbage's Secret: The Tale of a Cipher—and APL. Prentice Hall..

Quist-Aphetsi Kester: is a global award winner 2010 (First place Winner with Gold), in Canada Toronto, of the NSBE's Consulting Design Olympiad Awards and has been recognized as a Global Consulting Design Engineer. He is a PhD student in Computer Science. The PhD program is in collaboration between the AWBC/USFC Academics Without Borders/Universitaires Sans Frontieres (formerly AHED-Academics for Higher Education and Development) Canada and the Department of Computer Science and In-formation Technology (DCSIT), University of Cape Coast. He had a Master of Software Engineering degree from the OUM, Malaysia and BSC in Physics from the University of Cape Coast-UCC Ghana.

He has worked in various capacities as a peer reviewer for IEEE ICAST Conference, lecturer and Head of Comput-er science department. He is currently a lecturer and Head of Digital Forensic Laboratory Department at the Ghana Technology University.