Cryptography: A New Approach of Classical Hill Cipher

M. Nordin A. Rahman, A. F. A. Abidin, Mohd Kamir Yusof, N. S. M. Usop

Universiti Sultan Zainal Abidin, Terengganu, Malaysia
mohdnabd@unisza.edu.my, mohdkamir@unisza.edu.my

Abstract

The Hill cipher is the first polygraph cipher which has some advantages in symmetric data encryption. However, it is vulnerable to known plaintext attack. Another setback is that an invertible key matrix is needed for decryption and it is not suitable for encrypting a plaintext consisting of zeroes. The objective of this work is to modify the existing Hill cipher to overcome these three issues. Studies on previous results showed that the existing Hill algorithms are not yet sufficient. Some of these algorithms are still vulnerable to known plaintext attack. On the other hand, some of these algorithms have better randomization properties and as a result they are more resistant against known plaintext attack. Nevertheless, these enhanced Hill cipher algorithms still face the non invertible key matrix problem. Moreover, neither of these algorithms are suitable for all zeroes plaintext block encryption. In this paper, a robust Hill algorithm (Hill++) is proposed. The algorithm is an extension of the Affine Hill cipher. A random matrix key is introduced as an extra key for encryption. Moreover, an involuntary matrix key formulation is also implemented in the proposed algorithm. This formulation can produce an involuntary key where a same key can be used for both encryption and decryption. Testing on the proposed algorithm is carried out via two approaches, that is through comparative study and statistical analysis. Comparative study shows that Hill++ is resistant to all zeroes plaintext block encryption and does not face the non invertible key matrix problem as what was faced by the original Hill, AdvHill and HillMRIV algorithms. Apart from this, the encryption quality of the proposed algorithm is also measured by using the maximum deviation and correlation coefficient factors. Results from statistical analysis shows that Hill++ (when compared to Hill, AdvHill and HillMRIV algorithms) has the greatest maximum deviation value and its correlation coefficient value is the closest to zero. The results from these two measures proved that Hill++ has better encryption quality compared to HillMRIV.

Keywords: *Hill cipher, invertible key matrix, involuntary key, symmetric encryption*

1. Introduction

Today, information is one of the most valuable intangible assets. Due to this fact, information security has become an important issue. Cryptography is one of the methods to ensure confidentiality and integrity of information. It is from the Greek word “kryptos” which means hidden [1]. Cryptography is the art and science of making message unintelligible. It serves as a secret communication mechanism and can be traced back till thousands of years ago. Caesar’s cipher is one of the earliest known cryptosystem which was used by Julius Caesar to convey secret messages to Marcus Cicero. There are also other conventional ciphers such as the Hill cipher, the Vigenère cipher and the Affine cipher. All of these ciphers are the foundation for modern cryptography. There are two types of cryptosystems. They are the symmetric cryptosystem and the asymmetric cryptosystem. In symmetric cryptosystem, the sender and recipient share the same key. It means the same key is used for encryption and

decryption. In asymmetric cryptosystem, different keys are used. A public key is used by sender to encrypt the message while the recipient used a private key to decrypt it. In modern cryptographic implementation both asymmetric and symmetric cryptosystem are applied together. In this paper, we focus on one of the classical cipher mentioned earlier – the Hill cipher. Although its vulnerability to cryptanalysis has rendered it unusable in practice, it still serves as an important pedagogical role in cryptology and linear algebra [2]. Thus, a modified version of the Hill cipher will be designed to overcome the existing problems found in the cipher. In Section 2 we will discuss the Hill cipher. We continue with discussing previous work enhancing the Hill cipher in Section 3. Next, we discuss our proposed algorithm in Section 4. Our empirical analysis results will be presented in Section 5. Finally, we conclude in Section 6.

2. Hill Cipher

Hill cipher was first described in 1929 by its inventor, a mathematician Lester S. Hill, in the journal The American Mathematical Monthly [3]. Hill cipher is the first polygraphic cipher. A polygraphic cipher is a cipher where the plaintext is divided into groups of adjacent latters of the same fixed length *n*, and then each such group is transformed into a different group of *n* letters [3]. This polygraphic feature increased the speed and throughput of Hill cipher. Besides, it has some other advantages in data encryption such as its resistance to frequency analysis. The core of Hill cipher is matrix manipulation [4]. Its linear algebra equation is $C = K \times P \pmod{m}$, where C represents the ciphertext block, P represents the plaintext block and K is the key. The key, K is in the form of matrix. Thus, for decryption, an inverse key matrix, K^{-1} is needed.

3. Previous Work

Ismail, *et al.*, [5] proposed a modified Hill cipher which uses a unity (one-by-one) matrix as a key to encrypt each plaintext blocks. In this algorithm, each plaintext block is encrypted by using its own key. It is aimed to overcome the security flaw of the original Hill cipher where the same key matrix is used to encrypt all the plaintext blocks. To compute a unique key for each plaintext blocks, a secret initial vector, *IV* is needed. This *IV* is then multiplied with a randomly selected initial key and the multiplication results will be a unique key which can be used for encryption. Since the *IV* multiplication is performed row by row, this algorithm is known as HillMRIV (abbreviation for **H**ill **m**ultiplying **r**ows by **i**nitial **v**ector).

Research done by Rangel-Romero, *et al.*, [6] shown tha the algorithm proposed by Ismail et al. has a few major drawbacks which are similar with the original Hill cipher. Rangel-Romero, *et al.*, proved that the proposed algorithm is still vulnerable towards known plaintext attacks. Assume that the key, K_i used for encryption is a 2×2 key matrix and the initial vector , *IV* = [e, f]. In symmetric cryptography, attackers can easily get the encryption key since it is common practice to encrypt several plaintext blocks with a same key [6]. Assume that the attacker has successfully obtained the 2×2 matrix key. With this key, it is possible to calculate the *IV* values. Apart from its vulnerability to known plaintext attack, Rangel-Romero, *et al.*, also discussed some other drawbacks in Ismail, *et al.*'s algorithm. First of all, the algorithm is not suitable for all zeroes plaintext block encryption. An all zeroes plaintext block is a matrix block where all the values in it are zero. An example of a 2×2 all zeroes matrix block is $\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$. Usually, this will happen when Hill cipher is used to encrypt an image which a large portions of pixels in black. Note here that black pixels are mapped to zero in the standard grayscale image. Since the Hill cipher's linear algebra equation is $C = KP \pmod{m}$,

if C equals zero, P will equal to zero too. This result will cause the failure of Hill cipher in hiding the content of the image. Ismail et al. described that the *IV* can be randomly generated. However, not every matrix key computed from the random *IV* is invertible. If the key generated has no inverse, the decryption operation cannot be performed.

Rushdi, *et al.*, [7] also noticed the problem of non invertible matrix key in Hill cipher. Thus, they designed a robust cryptosystem algorithm for non invertible matrices. The non invertible matrix key problem is solved by converting each plaintext character into two ciphertext characters. So with the decryption, the process involved the conversion of two ciphertext characters into one plaintext character. Although this algorithm solved the non invertible matrix key problem, there are other problems which caused the algorithm not suitable to be implemented. In Rushdi, *et al.*'s algorithm, it needs to determine whether the key matrix's determinant is zero. Note here that a matrix with determinant zero cannot be inversed. However, this process of checking will increase the computational complexity and computational duration. Another drawback of this algorithm is transmission of 2*n* characters are required to encrypt *n* characters. Although the author claimed that nowadays we have very large capabilities of data transmission, it may also be problematic if there is low bandwidth.

Toorani, *et al.*, [2] created a variant of Hill cipher which is an extension of the affine Hill cipher. Affine Hill cipher is the combination of Hill cipher and the affine cipher. The affine Hill cipher is expressed in the form of C = PK + V (mod m) where V represents a constant in the form of matrix [8]. The algorithm proposed by Toorani, *et al.*, has the same structure like an affine Hill cipher. In this algorithm, each plaintext block is encrypted using a random number. It will increase the randomization of the algorithm and thus increased its strength towards common attacks. This algorithm is also aimed to avoid multiple random number generation. Thus, only one random number is generated at the beginning of encryption. A one-way hash function is used to generate the corresponding random number recursively. The random number is then used to compute *V*. The idea to compute *V* came from the Menezes-Qu-Vanstone (MQV) key-exchange protocol. The computation of *V* enhanced the resistance of the cipher towards known plaintext attack as *n* equations cannot be used for solving an unknown $n \times n$ matrix and 2*n* unknown parameters [2]. Toorani, *et al.*, also introduced a one-pass protocol for the sender and receiver to share the core random number. The advantage of one-pass protocol is its security remains even though it does not require any explicit authentication step. The algorithm is tested in term of computational costs. Results show that the proposed algorithm is computationally efficient. Although the algorithm proposed by Toorani, *et al.*, increased the randomization of Hill cipher, it still suffers the same problem as Ismail, *et al.*'s algorithm. The random number generated has the risk to produce a non invertible matrix key.

The problem of non invertible matrix key is solved by Bibhudendra, *et al.*, [9]. Bibhudendra, *et al.*, proposed a novel advanced Hill (AdvHill) which involved an involutory matrix key in its encryption algorithm. When an involutory key is used in encryption, the same key can be used for both encryption and decryption. Obviously, it reduced the computational complexity as the process of finding inverse key can be eliminated. Besides, different involutory key matrices can be used to encrypt different plaintext blocks. Although the $\frac{n}{2} \times \frac{n}{2}$ matrix is generated randomly, the algorithm is still able to produce a $n \times n$ involutory key. Thus, this algorithm can increase the randomization of the cipher without having the non invertible matrix key problem. This algorithm is used to encrypt both grayscale and color images. Results show that the proposed algorithm is more efficient when compared to the original Hill cipher. However, this algorithm does not overcome the issue mentioned by Rangel-Romero, *et al.*, where the Hill cipher is not suitable to encrypt all zeroes plaintext block.

In another paper, Bibhudendra, *et al.*, [4] proposed several solutions to overcome the non

invertible matrix key problem in Hill cipher. There are a total of three solutions proposed. These are the invertible matrix formulation, involutory matrix formulation and permutation matrix formulation. The invertible matrix formulation has one disadvantage. A checking operation is required to determine whether the matrix is singular. It is time consuming when it involves a high dimensional matrix. Due to this, the involutory matrix formulation is designed. In this method, the involutory matrix formulation will compute an involutory key. This key can be used for both encryption and decryption process. In comparison, the algorithm to produce this key is easier compared to previous method. No checking operation is required. The permutation matrix formulation is another method which is an extension of involutory matrix formulation. Different keys are used to encrypt each plaintext block. Different keys are generated based on the random permutations of columns and rows of a matrix. An $n \times n$ matrix can generate $n!$ number of involutory matrix. Although different keys are used for encryption, once the main matrix key which was used to generate the other keys is somehow compromised, then the message encrypted can easily cracked.

4. Proposed Algorithm

A. Mathematical Notation

We begin by presenting the notations used to describe our algorithm. The notations are as followed:

C = ciphertext
P = plaintext
K = n × n matrix key
RMK = random n × n matrix key
n = length of matrix row and column
m = value of modulus, m > 1
b = number of plaintext blocks
α_i = 1st seed number
β = 2nd seed number
γi = multiplying factor
where αi, β, γi ε Zp and i mod 2 = 0

B. Modular Arithmetic

For the benefit of the reader, we present here the relevant properties of the modular arithmetic utilized within the scope of our algorithm.

Let c ε Z. The following equations describe the modular arithmetic operations [4].

Addition:
$$(a + b) \bmod c = [(a \bmod p) + (b \bmod p)] \bmod c$$

Negation:
$$-a \bmod c = c - (a \bmod c)$$

Subtraction:
$$(a - b) \bmod c = [(a \bmod c) - (b \bmod c)] \bmod c$$

Multiplication:
$(a \times b) \bmod c = [(a \bmod c) \times (b \bmod c)] \bmod c$

Division:
$(a \div b) \bmod c = d$ when $a = (b \times d) \bmod c$

The following describe the algebraic property of the modular arithmetic. Further discussion on the algebraic properties of the modular arithmetic can be found in any modern algebra text.

Commutative Law:
$(a + b) \bmod c = (b + a) \bmod c$
$(a \times b) \bmod c = (b \times a) \bmod c$

Association Law:
$[(a + b) + d] \bmod c = [a + (b + d)] \bmod c$

Distribution Law:
$[a \times (b + d)] \bmod c =$
$[\{(a \times b) \bmod c\} + \{(a \times d) \bmod c\}] \bmod c$

Identities:
$(0 + a) \bmod c = a \bmod c$
$(1 \times a) \bmod c = a \bmod c$

C. Proposed Algorithm

The proposed algorithm is based on the affine Hill cipher. The affine Hill cipher is the combination of the affine cipher and Hill cipher. Differing from the Hill cipher, the plaintext is encrypted as $C = PK + V \pmod m$ where V represents a constant in the form of matrix. To produce a robust and reliable cryptosystem, we enhance this encryption core. This enhancement will solve the non invertible matrix key problem and increase the randomization of the algorithm and as a result increases its resistance towards known-plaintext attack. With the proposed algorithm, the plaintext is encrypted as $C = PK + RMK \pmod m$ with three parameters α, β and γ. These three parameters act as the secret keys which are shared between the sender and receiver.

The encryption algorithm is as followed:

1. Randomly select α_i, β_1 and γ_i. The range of the numbers which can be selected is within Z_m, where m is value of modulus. m must be a prime number. For instance, if m is 29, then the number which can be chosen is from 0 to 28. The sender is the one who responsible to determine the random elements α_i, β_1 and γ_i. These elements will be shared with the receiver to make the decryption possible.
2. Generate a set of pseudorandom numbers $\{x_1, x_2, \ldots, x_n\}$ based on α_i. The values α_i acts as a seed number for the pseudorandom number generator to generate the set of pseudorandom. The pseudorandom number generation function is supported by most of the programming language such as C, C++ and Java via srand() and rand().
3. Produce a $\frac{n}{2} \times \frac{n}{2}$ matrix based on the set of pseudorandom numbers $\{x_1, x_2, \ldots, x_n\}$. For instance, if $n = 4$, the $\frac{n}{2} \times \frac{n}{2}$ matrix is $\begin{bmatrix} x_1 & x_2 \\ x_3 & x_4 \end{bmatrix}$. This $\frac{n}{2} \times \frac{n}{2}$ matrix is vital as the

upcoming step will need a $\frac{n}{2} \times \frac{n}{2}$ matrix to generate an involutory matrix K_i.

4. With this $\frac{n}{2} \times \frac{n}{2}$ matrix, generate a n × n involutory matrix Ki based on the algorithm proposed by Bibhudendra et al. [4].
5. Generate a set of pseudorandom numbers $\{y_{11}, y_{12}, \ldots, y_{1n}, y_{21}, y_{22}, \ldots, y_{2n}, \ldots, y_{m1}, y_{m2}, \ldots, y_{mn}, \ldots, y_{n1}, y_{n2}, \ldots, y_{n\times n}\}$ based on β. The mechanism of pseudorandom number generation is similar with Step 2.
6. Produce a n × n matrix, $Ktemp_1$, based on the set of numbers $\{y_{11}, y_{12}, \ldots, y_{1n}, y_{21}, y_{22}, \ldots, y_{2n}, \ldots, y_{m1}, y_{m2}, \ldots, y_{mn}, \ldots, y_{n1}, y_{n2}, \ldots, y_{n\times n}\}$. For instance, if $n = 4$, the $Ktemp_1$ will be $\begin{bmatrix} y_{11} & \cdots & y_{1n} \\ \cdots & \cdots & \cdots \\ y_{n1} & \cdots & y_{nn} \end{bmatrix}$. The value $Ktemp_1$ will act as the foundation to compute RMK_1.
7. Compute RMK_1 by multiplying each rows of $Ktemp_1$ matrix key with γ_1. After the rows multiplication, RMK_1 becomes $\begin{bmatrix} y_{11}\gamma_1 & \cdots & y_{1n}\gamma_1 \\ \cdots & \cdots & \cdots \\ y_{n1}\gamma_1 & \cdots & y_{nn}\gamma_1 \end{bmatrix}$.

For the first ciphertext block C1, the encryption formula is:

$$C_1 = P_1K_1 + RMK_1 \pmod{m}. \quad (1)$$

Repeat Step 1 to 4 to produce the remaining involutory matrix key, $\{K_2, K_3, \ldots, K_b\}$. To compute the remaining n × n random matrix key, $\{RMK_2, RMK_3, \ldots, RMK_b\}$, it is given by the relation $RMK_i = C_{i-1} \times \gamma_i$, where $i = 2, 3, \ldots, b$. As an illustration, let $C_{i-1} = \begin{bmatrix} z_{11} & \cdots & z_{1n} \\ \cdots & \cdots & \cdots \\ z_{n1} & \cdots & z_{nn} \end{bmatrix}$ where C_{i-1} is a $n \times n$ ciphertext block. Then RMK_i after the rows multiplication becomes $\begin{bmatrix} z_{11}\gamma_1 & \cdots & z_{1n}\gamma_1 \\ \cdots & \cdots & \cdots \\ z_{n1}\gamma_1 & \cdots & z_{nn}\gamma_1 \end{bmatrix}$.

The new general algorithm for encryption is as follows:

$$C_i = \begin{cases} C_i = P_iK_i + RMK_i \pmod{m},\ i = 1 \\ C_i = P_iK_i + RMK_i \pmod{m} \\ \text{where } RMK_i = C_{i-1} \times \gamma_i,\ i = 2, 3, \ldots, b \end{cases} \quad (2)$$

The decryption algorithm is as follows:

Assume the receiver had obtained α_i, β and γ_i which are randomly selected by the sender. The mechanism of pseudorandom numbers and random matrix key generation for decryption is similar with encryption.

1. Generate $\{x_1, x_2, \ldots, x_n\}$ based on α_i.
2. Produce a $\frac{n}{2} \times \frac{n}{2}$ matrix based on the set of numbers $\{x_1, x_2, \ldots, x_n\}$.
3. With this $\frac{n}{2} \times \frac{n}{2}$ matrix, generate an involutory matrix K_i based on the algorithm proposed by Bibhudendra et al. [4].

4. Generate $\{y_1, y_2, \ldots, y_{n\times n}\}$ based on β_1.
5. Produce a $n \times n$ matrix, Ktemp_1 based on the set of numbers $\{y_1, y_2, \ldots, y_{n\times n}\}$.
6. Compute RMK_1 by multiplying each rows of Ktemp1 matrix key with γ_1.

For the first plaintext block P_1, the decryption formula is:

$$P_1 = (C_1 - \text{RMK}_1) \times K_1 \pmod{m} \quad (3)$$

To compute the remaining n × n random matrix key, $\{\text{RMK}_2, \text{RMK}_3, \ldots, \text{RMK}_b\}$, it is given by the relation $\text{RMK}_i = C_{i-1} \times \gamma_i$, where $i = 2, 3, \ldots, b$.

The new general algorithm for decryption is as follows:

$$P_i = \begin{cases} P_i = (C_i - \text{RMK}_i) \times K_i \pmod{m},\ i = 1 \\ P_i = (C_i - \text{RMK}_i) \times K_i \pmod{m} \\ \text{where } \text{RMK}_i = C_{i-1} \times \gamma_i,\ i = 2, 3, \ldots, b \end{cases} \quad (4)$$

The proposed algorithm enhanced the concept of affine Hill cipher instead of the original Hill cipher. The Affine Hill cipher can solve some of the security flaws in the original Hill cipher. Recap the issue raised up by Tangel-Romero, *et al.*, [6]. The original Hill cipher will fail to hide any information if there are all zeroes plaintext block. It is because the Hill cipher encryption formula is C = KP (mod *m*). Thus, if C equals zero, P equals zero too. However, this problem will not happen to the affine Hill cipher since if C equal zero, P will equal RMK, which is not zero.

In the proposed algorithm all keys are constructed based on different sets of pseudorandom numbers (Step 2, 3, 5 and 6 in encryption). These pseudorandom numbers are generated from three seed numbers, which are α_i, β and γ_i. It is aimed to increase the randomization of the algorithm. Randomization is important for a cipher as better randomization can yield better encryption quality.

The computation of RMK (Step 5, 6 and 7 in encryption) takes advantages of ideas behind HillMRIV [19]. HillMRIV used an initial vector to multiply with a key matrix to produce the remaining keys. If it is an $n \times n$ matrix, then there will be $n!$ numbers of initial vectors. In the proposed algorithm, the initial vector is known as multiplying factor. Instead of using the set of pseudorandom numbers to construct RMK, a multiplying factor is added (Step 7 in encryption) as it can enhance the security of the proposed algorithm. Note here that if γ_i is somehow compromised, the attacker can generate the RMK based on this seed number. The function of multiplying factor is to increase the resistance of the proposed algorithm towards attacks as an attacker will need both β and γ_i to compute RMK. However, instead of using $n!$ numbers of multiplying factor, only one multiplying factor is used. We state the following conjecture:

Conjecture 4.1

Let Hill++ algorithms utilize only one multiplying factor. Randomization provided is equivalent to Hill++ utilizing $n!$ multiplying factors.

The rationale of using only one multiplying factor is to reduce the computational complexity. Observe that only RMK_1 is produces based on the seed number, β which generates a $n \times n$ key matrix. The remaining sequence $\{\text{RMK}_2, \text{RMK}_3, \ldots, \text{RMK}_b\}$ are

produced by multiplying the multiplying factor, γ_i with the preceding ciphertext blocks, C_{i-1}. Since thr first RMK (i.e. RMK_1) is needed to encrypt the first plaintext block and encryption has no reference to any preceding ciphertext blocks, the seed number, β is needed. The combination of preceding ciphertext blocks and the multiplying factor enhance the resistance of the algorithm towards known plaintext attack. Since RMK changes over every plaintext blocks, the number of unknowns become more that the number of equation available for an attacker [7]. If β and γ_i are somehow compromised, the attacker is still unable to crack the encrypted message.

5. Experimental Result and Analysis

Experiment of the proposed algorithm is carried out via two approaches, which are cipher comparison and measurement of encryption's quality. Table 1 shows that Hill++ has a few features which obviously overcome some of the vulnerabilities in the existing Hill algorithms.

Table 1. Comparison Between Hill++ And Existing Algorithms

Cipher	Comparison Factor	
	Need Inverse Key Matrix	Vulnerable if there are all zeroes blocks
Original Hill	Yes	Yes
Ismail, *et al.*'s Algorithm	Yes	Yes
Rushdi, *et al.*'s Algorithm	Yes	Yes
Bibhudendra, *et al.*'s Algorithm	No	Yes
Hill++	No	No

One can observe from Table 1 that Hill++ is better than all the previous Hill algorithm as it fulfills the two comparison factors. However, this is not yet enough to judge the encryption quality of the proposed algorithm. Thus, two measuring techniques have been used to evaluate the encryption quality of the proposed algorithm. These techniques have been used to evaluate the encryption quality of the proposed algorithm. These techniques are the maximum deviation measures [10] and the correlation coefficient measures.

A. The Maximum Deviation Factor

This measurement technique measures the quality of encryption based on the deviation between the plaintext and ciphertext. The more the ciphertext is deviated from the plaintext, the better is the encryption algorithm. The steps involved in the measurement process are as follows [22]:

1. Count the number of characters (in numerical representation) from 0 to p-1 (based on p). Base on the results, construct a graph (in the form of curve or histogram) for both the plaintext and the ciphertext.
2. Construct another graph which will show the absolute difference between the plaintext and the ciphertext. Compute the sum of deviations, D between the plaintext graph and the ciphertext graph. This can be done by the trapezoidal rule:

$$D = \frac{h_0 + h_{p-1}}{2} + \sum_{i=1}^{p-1} h_i \qquad (5)$$

h_i is the amplitude of the absolute difference curve at value *i*. Since D is the sum of deviations, thus, the higher the value of D, the better is the quality of the encryption.

B. The Correlation Coefficient Factor

This measurement technique measures the quality of encryption based on the relationship between two variables, which are plaintext and ciphertext in this case. If the correlation coefficient is one, it means that the plaintext and ciphertext are highly dependent. If the correlation coefficient is one, it means that the plaintext and ciphertext are highly dependent. If the correlation coefficient is zero, it means that the ciphertext and the plaintext are not correlated (these two variables are independent between each other). Thus, the smaller is the values of the correlation coefficient; the better is the quality of encryption. The correlation coefficient (CC) can be computed by using the following equation:

$$CC = \frac{\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y))}{\sqrt{\sum_{i=1}^{N}(x_i - E(x))^2}\sqrt{\sum_{i=1}^{N}(y_i - E(y))^2}} \qquad (6)$$

where $E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i$, *x* and y are the numerical representation of the plaintext and ciphertext.

C. Example Test Case

To obtain the results based on the two measurement techniques mentioned previously, a test case is needed. Thus, a simple message is designed. This message consists of one hundred and twelve characters. Since the matrix block is a 4×4 matrix, the length of message is fixed to one hundred and twelve so that there are a total of seven matrix blocks. The message is "*general ahmad, launch the missile on first october two thousand ten at eleven ocolock in the morning.from aizatmod*". We first assign the characters in the test case to numerical values. Table 2 shows the numerical representation of twenty nine characters which will be used in the cipher encryption and decryption. This twenty nine characters includes small letters of the alphabet from a to z, commas (,) and blank space (␣). Note here that the length of characters is not fixed to twenty nine. This length of characters is just for the demonstration purposed. Most research used either twenty six or twenty nine characters.

Table 2. Numerical Representation of Twenty Nine Characters

a	b	c	d	e	f	g	h	i	J
0	1	2	3	4	5	6	7	8	9
k	l	m	n	o	p	q	r	s	T
10	11	12	13	14	15	16	17	18	19
u	v	w	x	y	z	.	,	␣	
20	21	22	23	24	25	26	27	28	

Based on the test case and the numerical representation table, the message is encrypted by using three different Hill cipher algorithm. These algorithms are the original Hill, HillMRIV and Hill++. The encrypted messages are presented in Table 3.

After encryption, the encrypted messages are first converted to numerical values. Based on the numerical values, graphs are constructed to show the deviation to the numbers of all characters between plaintext and the corresponding ciphertext. Figure 1 shows deviation graph between plaintext, encrypting with the original Hill and encrypting with Hill++. While in Figure 2 it shows the deviation graph between plaintext, encrypting with HillMRIV and encrypting with Hill++. The x-axis in both the graphs represents the characters in numerical values while the y-axis represents the number of each character that appears in the ciphertext.

Table 3. Encrypted Message Using Original Hill, Hill MRIV and Hill++

Cipher	Encrypted Message
Original Hill	,h altyezdndfdtkv,xhxkyemf.pxliewkqrzwst . ht.l,d kwdqqfnxpkcfh,tt hrvnnytsat,nfs.m,n.jqt dlnonvqb ,xrpllesnorau,o
HillMRIV	,h altyezdndfdtkfpeolmnihgcbltoiy.ukwfisu,g . ,vvx iby.hurmmlqtxvhcgultjehciaomjg,zstkz w st.kjfstei upcocyogvpah..
Hill++	tzejfmrdzh arnljz.snlyvocvplbasjffltttwpvofvpknb sioyoffvjdqksb g .fqzofg.,nzpvdoyssz vmfkyhjakd baprzsag.ak.lgf.w

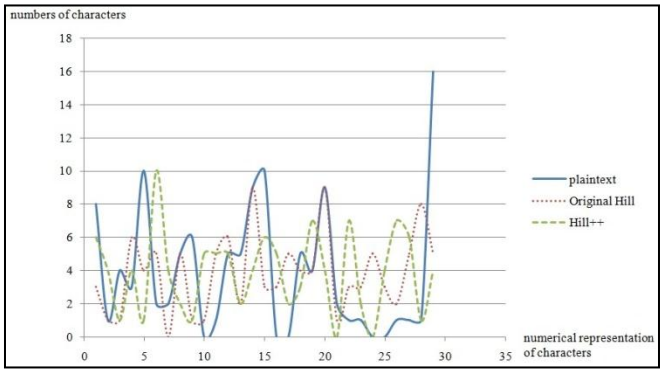

Figure 1. Deviation Graph between Plaintext, Original Hill Cipher and Hill++

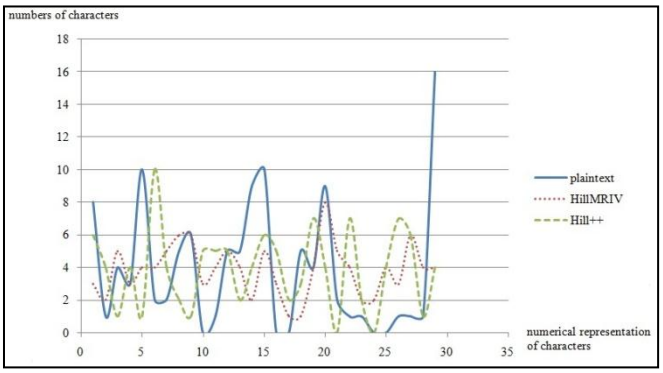

Figure 2. Deviation Graph between Plaintext, HillMRIV and Hill++

Based on the graphs in Figure 1 and Figure 2, another graph is constructed (refer to Figure 3). This graph shows three curves which represent the absolute difference between plaintext and ciphertext (encrypted by using three different algorithms; the original Hill, HillMRIV and Hill++ respectively). The wider is the area under the absolute difference curves, the better is the quality of encryption. It is because the more the ciphertext values deviated from the plaintext values, the better is the quality of encryption. Note here that all graphs in this section are computed with the assistance of Microsoft Excel 2007.

Finally in Table 4, we show the comparison of the experiment results from two different measuring factors. In Table 4, MF1 indicates the maximum deviation measures while MF2 represent the correlation coefficient measure. MF1 is computed based on the sum of deviation (area) from the graph in Figure 3.

As a note a negative value of correlation coefficient means that the two variables (plaintext and ciphertext) are negatively correlated. By Table 4 we can observe that Hill++ produces MF1 value which larger than the values produces by the other algorithms. It also produces a value of MF2 which is the nearest to zero. Hence, it can be concluded that Hill++ is a better algorithm compared to the original Hill and HillMRIV when based upon the two properties tested here.

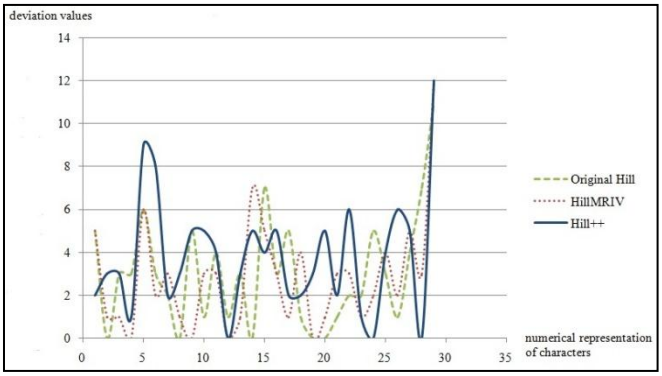

Figure 3. Absolute difference curve for original Hill Cipher, HillMRIV and Hill++

Table 4. Evaluation of Encryption Quality for Original Hill Cipher, HillMRIV and Hill++

Cipher	Measuring Factor	
	MF1	MF2
Original Hill	80	0.07553
HillMRIV	73.5	0.06059
Hill++	103	-0.00081

6. Conclusion

We have presented a modified version of Hill cipher which is an extension of the affine Hill cipher, known as Hill++. Hill++ introduces a random matrix key which is computed based on the previous ciphertext blocks and a multiplying factor. This significantly increased the resistance of the algorithm to the known plaintext attack. Hill++ also implements an involutory key generation algorithm by Bibhudendra et al. where the same matrix key can be used for both encryption and decryption. As a consequence Hill++ does not required any additional operation to compute the inverse matrix key, which is definitely more time-saving. By comparing experimental results it shows that Hill++ is the only algorithm which fulfills both comparison factors, does not need an inverse matrix key ad is not vulnerable when encrypting all zeroes plaintext block. Statistical analysis presented also shows satisfactory results. Hill++ has better encryption quality compared to the original Hill cipher and HillMRIV.

Acknowledgements

Special thanks to Universiti Sultan Zainal Abidin (UniSZA) my friend Che Mat Ismail for his support and advice.

References

[1] D. Luciano and G. Prichett, “From Caesar Ciphers to Public-Key Cryptosystem”, The College Mathematics Journal, vol. 12, no. 1, **(1987)**, pp. 2-17.
[2] M. Toorani and A. Falahati, “A Secure Variant of the Hill Cipher”, in Proc. 14th IEEE Symposium on Computers and Communications, Sousse, **(2009)**, pp. 313-316.
[3] M. Eisenberg, “Hill ciphers and modular linear algebra”, Mimeographed notes, University of Massachusetts, **(1998)**.
[4] I. A. Ismail, M. Amin and H. Diab, “How to repair the Hill Cipher”, Journal of Zhejiang University Science A, vol. 7, no. 12, **(2006)**, pp. 2022-2030.
[5] A. Bibhudendra, “Novel Methods of Generating Self-Invertible Matrix for Hill Cipher Algorithm”, International Journal of Security, vol. 1, no. 1, **(2006)**, pp. 14-21.
[6] Y. Rangel-Romero, G. Vega-Garcı́a, A. Menchaca-Méndez, D. Acoltzi-Cervantes, L. Martı́nez-Ramos, M. Mecate-Zambrano, F. Montalvo-Lezama, J. Barrón-Vidales, N. Cortez-Duarte and F. Rodrı́guez-Henrı́quez, “Comments on How to repair the Hill cipher”, Journal of Zhejiang University Science A, vol. 9, no. 2, **(2006)**, pp. 211-214.
[7] A. H. Rushdi and F. Mousa, “Design of a Robust Cryptosystem Algorithm for Non-Invertible Matrices Based on Hill Cipher”, Int’l Journal of Computer Science and Network Security, vol. 9, no. 5, **(2009)**, pp. 11-16.
[8] D. R. Stinson, “Cryptography Theory and Practice”, 3rd edition. Chapman & Hall/CRC, **(2006)**, pp. 13-37.
[9] A. Bibhudendra, K. P. Saroj, K. P. Sarat and P. Ganapati, “Image Encryption using Advanced Hill Cipher Algorithm”, International Journal of Recent Trends in Engineering, vol. 1, no. 1, **(2009)**, pp. 663-667.
[10] I. E. Ziedan, M. M. Fouad and D. H. Salem, “Application of data encryption standard to bitmap and JPEG images”, in Proc. 12th National Radio Science conference, Cairo, **(2003)**, pp. 1-8.