Jurnal Ilmiah KURSOR
Menuju Solusi Teknologi Informasi
Vol. 8, No. 3, Juli 2016

ISSN 0216 – 0544
e-ISSN 2301– 6914

HILLMAIL: A SECURE EMAIL SYSTEM FOR ANDROID-BASED MOBILE PHONE USING HILL CIPHER ALGORITHM

[a]Tita Karlita, [b]Isbat Uzzin Nadhori, [c]Mohammad Azis Khoirul Fata
[a,b,c]Jurusan Teknik Informatika, Departemen Teknologi Informasi dan Komputer
Politeknik Elektronika Negeri Surabaya
Jl. Raya ITS - Kampus PENS Sukolilo
Surabaya 60111, INDONESIA
E-mail: tita@pens.ac.id

Abstract

Nowadays, email has become the most widely communication way in daily life. Email is a very important method of communicating across the internet. During transmission and downloads, it uses protocols which are not secure. Spammers and scammers misuse these protocols to gain access to critical data stored in the email. This triggered concerns because sometimes email is used to exchange confidential messages. To improve security and efficiency of email system, we made an email security system for Android mobile phone using Hill Cipher algorithm. Hill Cipher is a classic cryptography algorithm that uses matrix inverse and matrix multiplication operations to hide the message. The initial stage of the encryption process is forming ciphertext by multiplying the key matrix with plaintext matrix. The contents of encrypted messages only can be read by legitimate recipient who has the key. Converting the ciphertext into plaintext is done by multiplying the ciphertext matrix with the inverse key matrix. The email content can be a plain message or a message with an attached file.

Keywords: Android, cryptography, email, Hill Cipher, security.

INTRODUCTION

With rapid developments in communication technologies based on computer and internet, communications via emails has become more and more widespread. Other than computers, smartphones and tablets can no longer be ingnored by email users. Users often check their email using smartphone.

However, traditional email protocol is insecure since the message is transmitted in plain text. If someone wants to interpret, copy or even alter emails, they can do it with relative ease. There are many critical data sent via email. Individual privacies such as bank transactions, commercial secrets, even countries intelligence information are being delivered through emails and thus contents of emails are now more valuable than ever [1].

Therefore, the security of emails has raised more concerns. The secure messaging system has three benefits: keeping sensitive information private, preventing anyone from interfering with the contents of the message and authenticating the identity of both the message sender and receiver.

To overcome these problems, as a solution we implements cryptographic methods to email system to add security on the email system using Hill Cipher algorithms on the Android smartphone. Using this application, users can access, send and receive email messages safely and without fear of the message will be known to the parties who are not interested.

ORIGINALITY

Smartphone has now become a part of the world community, provides extraordinary ease to the process of inter-personal communication or groups as well as a media of information dissemination. But the internet media has loopholes that allow others to commit theft of data including via email, we need a tool that can manage data securely and efficiently so as not to overload the system.

This study proposes a new approach to build email security system uses Hill Cipher algorithms on android based mobile phone. This system works by doing cryptography on messages using Hill Cipher algorithms. Messages sent via the sender's android device and can only be opened by the same application on receiver's android device. From the experimental results it shows that by using Hill Cipher, encrypted message will have the same size as the original message. So it does not overload the system and secure, since the process of sending up to receive messages.

RELATED WORKS

Robinson proposes a Cryptography as a Service (CaaS) model, which allows operations to be performed via web services. The core value proposition is without having keys on a mobile device including send and receive signed and encrypted messages, view encrypted data stored on the phone and view encrypted data stored in the cloud [2]. Matrix manipulations is the principal works of Hill Cipher algorithm. For decryption, the inverse of matrix requires an inverse of the matrix that doesn't always exist. If the matrix is not invertible then encrypted text cannot be decrypted. Amin et al [3] use of self repetitive matrix. When this matrix multiplied with itself for a given mod value (i.e. mod value of the matrix is taken after every multiplication) will eventually result in an identity matrix after N multiplications. So, after N+ 1 multiplication the matrix will repeat itself.

Bodur and Kara implemented RSA encryption algorithm on the Short Message Service. The secure messaging process on the SMS channel are realized in the devices with the Android operating system [4]. A secure Short Message System using Hill Cipher algorithm also developed in [5] and the combined with compression using Arithmetic Coding algorithm in [6].

Encryption has been implemented in other media. Rahman implements Blowfish algorithm to secure email messages. This application is based on desktop, so it cannot be used by mobile users [7]. Acharya et al. encrypt an image using a technique different from the conventional Hill Cipher. A novel advanced Hill encryption technique has been proposed which uses an involutory key matrix. The scheme is a fast encryption scheme which overcomes problems of encrypting the images with homogeneous background [8].

HILL CIPHER ALGORITHM

Hill Cipher is a cryptography algorithm that implements modulo arithmetic. This cryptographic technique using a square matrix as a key used in the encryption and decryption process. Hill Cipher invented by Lester S. Hill in 1929. Hill Cipher not substitution method like other classical cryptography method, but using the matrix multiplication as the basis for encryption and decryption process.

In cryptography, *ciphertext* is the result of encryption performed on plaintext using an algorithm, called a cipher. *So, ciphertext* is encrypted text. Plaintext is what you have before encryption, and *ciphertext* is the encrypted result. On the Hill Cipher, plaintext divided into blocks of a certain size. Each character in a block of plaintext will affect the other characters in the process of encryption and decryption, so the same plaintext character will not produce the same character in the ciphertext, and produce same size between plaintext and chipertext

Basic Hill Cipher Methods is modulo matrix. In the implementation, Hill Cipher Method using matrix multiplication and the matrix inverse. Hill Cipher key is n x n matrix where n is the block size. If n = 2, then the encryption is done every two characters. Matrix K as the key must be an invertible matrix, which has an inverse K^{-1}, so that:

$$K.K^{-1} = I \quad (1)$$

The Key must have an inverse, because the matrix K^{-1} is a key used to decrypt it.

Hill Cipher Encryption Algorithm

The stages of Encryption Hill Cipher is as follows:

1. Create a matrix K as the key size n x n

$$K_{nxn} = \begin{pmatrix} K_{11} & K_{12} & \dots & K_{1n} \\ K_{21} & K_{22} & \dots & K_{2n} \\ \dots & \dots & \dots & \dots \\ K_{n1} & K_{n2} & \dots & K_{nn} \end{pmatrix} \quad (2)$$

2. Substitute alphabet with numeric code
   $A \rightarrow 1, B \rightarrow 2, C \rightarrow 3, \dots, Z \rightarrow 0$
3. Categorize numbers obtained into several blocks vector P whose length is equal to the size of the matrix K.
4. Calculate Ciphertext (modulo 26) for each vector P
   $$C = K.P(mod\ 26) \quad (3)$$
   C = Ciphertext
   K = Key Matrix
   P = Plaintext
5. Restore each number in a vector C to the letter using the phase 2 to get the ciphertext.

Hill Cipher Decryption Algorithm

The stages of decryption Hill Cipher is as follows:

1. Substitute alphabet with numeric code
   $A \rightarrow 1, B \rightarrow 2, C \rightarrow 3, \dots, Z \rightarrow 0$
2. The key used to decrypt the ciphertext into plaintext is the inverse matrix K_{nxn}
3. Calculate K^{-1} (invers) :
   $$K^{-1} = \frac{1}{Det\ K}\ adj(K) \quad (4)$$
4. Calculate plaintext :
   $$P = K^{-I}.C \quad (5)$$
5. Restore each number in a vector P to the letter using the phase 2 to get the plaintext.

RESEARCH METHODS

This section discusses the design of cryptography process which includes encryption and decryption for Android email client using Hill Cipher algorithm. Then, this application is called with HillMail. Hill Cipher is a classic cryptography algorithm that uses the matrix inverse and matrix multiplication operations to hide the message.

The initial stage is forming ciphertext by multiplying the key matrix with messages matrix, then returns the ciphertext into plaintext by multiplying the ciphertext matrix with the inverse key matrix.

HillMail System Architecture

HillMail is an independent application that runs on android smartphone as an email client application. HillMail acts as a liaison between the user and the email server. Sender sends a message to the email server and email server

Figure 1. HillMail System Architecture

sends a message to the recipient in accordance with the request of the recipient.

Figure 1 is a system architecture design between mail client and mail server. Through the application of HillMail, the sender (user 1) use the key to encrypt the message using Hill Cipher method. Furthermore, email is sent via SMTP port and protocol to the mail server.

In the mail server messages are stored in the form of ciphertext, so it cannot be read by others. Ciphertext then sent to the mail server destination. With IMAP protocol, receiver (user 2) obtains the email from mail servers.

At this stage, the recipient gets a message in the form of ciphertext that is still locked. In order to read the message, it should be returned to the plaintext. This is done by using the decryption method. The same key used by the sender, so the resulting messages readable (in the form of plaintext).

The main focus of this system is to secure the message and make sure the message just received and opened by the recipient using HillMail applications which have a key to open it.

Figure 2 illustrates the encryption process of HillMail application. The encryption process begins when the user entering a message called the plaintext and a key. The length of the key is at least 1 character and a maximum of 9 characters. This is because all the characters and the key messages that the user entered will be converted into the form of a matrix. Plaintext characters and key characters will fill the column matrices. Encryption process is done using Hill Cipher algorithms and then generates ciphertext. After forming the ciphertext, then the message which has been encrypted will be sent to the email server to the destination address.

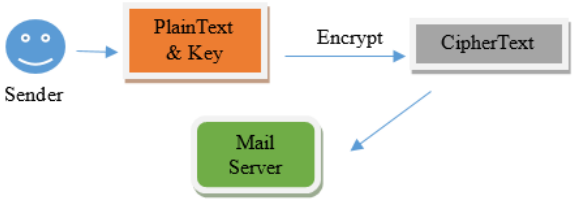

Figure 2. Encryption Process

Figure 3. Decryption Process

Figure 3 illustrates the decryption process using HillMail. After the ciphertext stored on the server, then the system will take it from server and decrypt it to generate plaintext that can be read by receiver.

Encryption and Decryption Using Hill Cipher Algorithm

Hill Cipher uses a square matrix as the key used for encryption and decryption. The process is done by multiplying plaintext matrix and key matrix that will produce ciphertext. Because the matrix operation is an arithmetic operation which requires numbers then the message that by default has String type need to be converted to numbers or integers. String conversion into numbers can be seen at Table 1.

Table 1 is an index table that contains a list of characters and the index number representing a total of 97 characters. Total index matrix must be a prime number, because to qualify Hill Cipher key generation which will be explained later in the section Key Generation. For example the word LAPAR means having the order index 21, 10, 25, 10, and 27. That numbers later will represent strings that are placed in the matrix.

Table 1. String conversion into numbers

0	1	2	3	4	5	6	7	8	9	A	B	C	D
0	1	2	3	4	5	6	7	8	9	10	11	12	13
E	F	G	H	I	J	K	L	M	N	O	P	Q	R
14	15	16	17	18	19	20	21	22	23	24	25	26	27
S	T	U	V	W	X	Y	Z	a	b	c	d	e	f
28	29	30	31	32	33	34	35	36	37	38	39	40	41
g	h	i	j	k	l	m	n	o	p	q	r	s	t
42	43	44	45	46	47	48	49	50	51	52	53	54	55
u	v	w	x	y	z		!	@	#	$	%	^	&
56	57	58	59	60	61	62	63	64	65	66	67	68	69
*	(	)	_	+	-	=	[	]	{	}	\	|	:
70	71	72	73	74	75	76	77	78	79	80	81	82	83
;	'	"	<	>	,	.	~	`	?	/	\t	\n	
84	85	86	87	88	89	90	91	92	93	94	95	96	

Key Generation

There are several requirements that must be met in the key generation using Hill Cipher algorithms:

1. Matrix key *n* x *n*

Must use the square key matrix *n* x *n* where *n* is the block size. The number of key characters that the user entered should be as much the result of a number of squares, for example, 4, 9, 16 and so on. To deal with restriction squared numbers in the key, the program is set to fulfill a key part which is missing if the user does not enter a key with a number of results numbers of squares. For example, if the user enters key words LAPAR by the number of 5 characters, the word is not eligible. Supposedly character length amounted to 4, 9 or 16. Then the program automatically adds the index string a number of shortcomings, namely three characters. Extra characters starting at index 0 of a list of strings and so on, so that the key has to be generated into LAPAR123.

2. Matrix Key Invertible

Matrix *K* which is the key in the encryption process must be an invertible matrix, which has the inverse K^{-1} and I is the identity matrix so that:

$$K.K^{-1} = I \quad (6)$$

Matrix will have an inverse if,

a. Determinant matrix is not equal to zero.
b. Does not have the same number factorization with an index number String Hill Cipher. To handle this problem, the number of index character strings has to be primes.

Once this is done, the key has been ready for use in the encryption or decryption process.

Encryption

The encryption process is done by multiplying the key matrix with the messages matrix. Then do the modulo operation between the result of multiplying and the total of String index. This is done so that the value of the matrix is not beyond index String. So the value of the matrix can be converted back into the form of String and can produce a sequence of characters of String called ciphertext.

Plaintext is grouped according to the value of n in the matrix nxn. Suppose there are plaintext MENCOBA ENKRIPSI and the value of n is 3, then plaintext grouped into threes as:

MEN|COB|A E|NKR|IPS|I

If the last character group has a number of characters is less than 3 it will be resurrected random characters to meet n x m matrix message. Then each group of characters multiplied by 3x3 matrix key. Furthermore, doing the modulo operation between the result of multiplying matrices and the number of index String. Results of modulo operation will be converted back to type String, and ciphertext can be formed. The message has been successfully encoded with algorithms Hill Cipher.

Decryption

Unlike the encryption process that produces ciphertext from plaintext matrix multiplication with matrix key, the decryption process to restore the plaintext to ciphertext is done by multiplying the plaintext matrix with the inverse of with the key matrix.

Here are the steps to produce the inverse matrix:

1. Calculate the minor matrix.
2. Then change to cofactor matrix.
3. Do adjoint matrix
4. Divide with the determinant

After inverse matrix key is created, the next step is to multiply the inverse of key matrix with ciphertext matrix. Once all steps have been passed it will produce the plaintext.

Mail Server Using JavaMail

JavaMail has two main functions, namely the function of sending messages and receiving messages. Sending a message using the SMTP protocol while receiving messages using POP or IMAP protocols. Here is a more detailed description of the application HillMail JavaMail implementation:

Sending message

1. Get the Session object from Mail Server. Here is the code snippets how to get Session object from Mail Server:

```
Properties props = new Properties();
props.put("mail.smtp.auth", "true");
props.put("mail.smtp.starttls.enable", "true");
props.put("mail.smtp.host", “smtp.gmail.com”);
props.put("mail.smtp.port", "465");

// Get the Session object.
Session session = Session.getInstance(props, new
javax.mail.Authenticator() {
    protected PasswordAuthentication
    getPasswordAuthentication() {
        return new PasswordAuthentication(username,
        password);
    }
});
```

Session is used to gain access to Mail server license. Without the Session, we will never connect to the Mail Server. In the session, host and port Mail server to be accessed has been defined. HillMail application using Google mail servers with SMTP host smtp.gmail.com with port 465. And Yahoo Mail server with a host smtp.mail.yahoo.com.

Furthermore, after the host is set, we are required to enter the username and password that have been registered in the email server to be accessed. It is used as user authentication to gain access rights to send mail using that account.

2. Compose a message, making the content of email messages to be sent, it includes input the recipient's email address, message subject, message body and attachments if any.
3. Send the message, used to send email messages to recipients. Email delivery using command Transport.send(message);.

Receiving message

To receive messages, HillMail app using the IMAP protocol. Here are some steps to get the message from the mail server:

1. Get a Session, the same as sending an email, receive email also need a session as user access rights to the mail server.
2. Making IMAP object store, and connect with store email server.
3. Creating an object folder and then open the inbox folder on the server
4. Then take all the messages of the inbox, downloaded all into the application. Then displayed in list view in the form of the sender's name and subject of the message contents.

Flagging

Flagging messages intended for the application to be able to distinguish whether the program will send regular email or encrypted email. Flag posted on the message is characters that are added at the last index of string message. Flag used is the following characters: #@Qj#. Signs are placed only on messages that are encrypted. This is done so that the application can provide the correct response to the message or attachment found in an email. If the message or attachments you have received in the form of the ciphertext, then the application will automatically offer to open the locked message, of course, by asking the correct key input from the recipient.

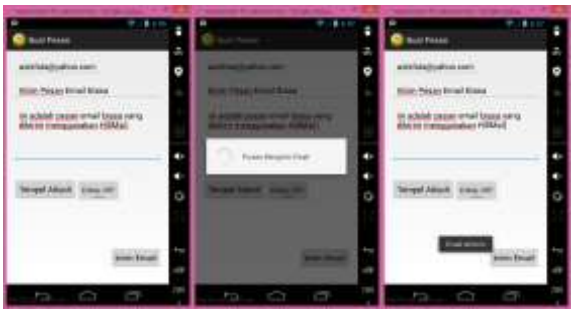

Figure 4. Sending Email

RESULT AND DISCUSSION

To run the HillMail application, we used gmail.com as a mail server. Starting from a sender sends a message until the message is received by the recipient. Here is the steps of testing the HillMail application:

Login

The purpose of this process is to ensure that the login process on the email account successfully done so can do sending and receiving messages. The account used to login further used for authentication when sending email and retrieve email from the mail server. Once logged in, then the Send and Retrieve mail can be done.

Sending Email

The data should be included in the process of sending message is the address of the recipient, subject, and message body. If we want to attach a file then the file can be attached by pressing "Attach Paste". Next press the Encrypt button to encrypt the message and click "Send Email". Figure 4 illustrates the steps to send an email without attaching a file.

Retrieve Email from Mail Server

The purpose of this test is to ensure that the HillMail email client can retrieve email from Gmail server. When the “Lihat” button in the inbox is pressed, then automatically HillMail will be connected to the server and perform authentication in order to take out all the contents of the message in the inbox folder on the mail server. Retrieving messages using the IMAP protocol which the application can take some information from email. Figure 5 is the result of retrieving email. It is shown in the figure 5 that information has displayed in the window are a list of senders and subject message.

Email Encryption Without Attaching File

The purpose of this test is to ensure that the application HillMail encrypts email messages. Figure 6 illustrates sender window when sending message without attaching file. Figure 7 illustrates recipients window when opening message.

In figure 6 and 7 is seen that the process is changing plaintext into encoded text. It appears that the ciphertext is made up of random characters so that will not be able to read its meaning.

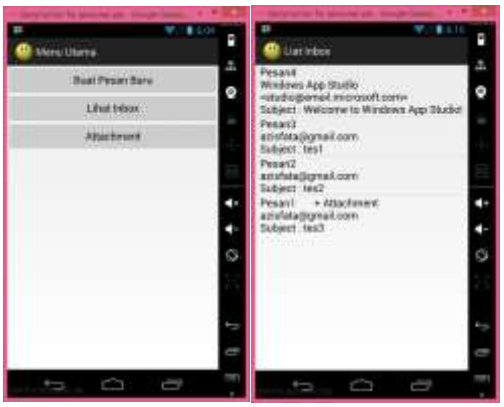

Figure 5. Retrieve Email

Figure 6. Email encryption without attaching file

Figure 7. Message window seen by recipients

To send an encrypted message, character message should be wrapped in a .txt file. This is because there are some characters from encryption that cannot be read by the mail server, so there will be some encoded characters are missing. To address them, the ciphertext must be sent as a file attachment so

that the integrity of the character ciphertext can be maintained.

By not considering the character flag, that is #@Qj#, the number of characters in the plaintext has an equal number to the number of characters in the ciphertext. From the results of this experiment can be concluded that the encryption algorithms using Hill Cipher produce ciphertext with the same size as the plaintext. So it does not increase the file size of the message. Email server can also receive encrypted messages such well, with no message content is missing.

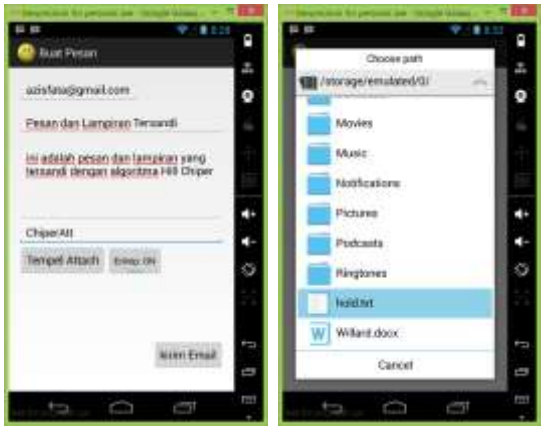

Figure 8. Sender writes message with file attachment

Email Encryption With Attaching File

This process is to send a message and attach a file containing text. For this case, the encryption is done either in the message or attachment. The same encryption technique is done both on the attachment file and the message. Attachment content is read first, then do the encryption using the same keywords with keywords that are used for message encryption.

Figure 8 illustrates the window when sender writes message, enter a key, and then add file attachment. Figure 9 shows the option wheter the sender want to encrypt the attachment file or not. Figure 10 shows the ciphertext of message and attachment file.

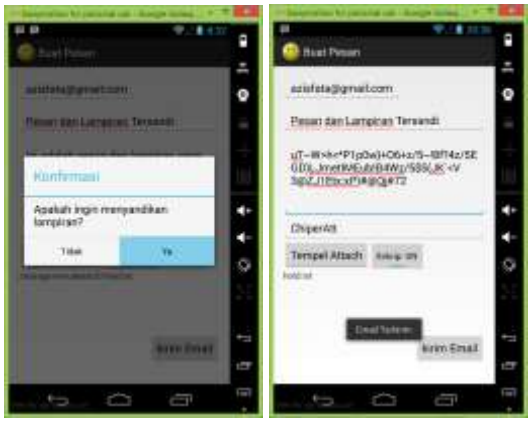

Figure 9. Encrypt the message and produce ciphertext

Email Decryption

This process is used to decrypt the message. The first step is to read the text in a file attachment that contains encrypted messages. Then the text is decrypted by inserting the key. If the key is correct, the original message (plaintext) will be seen. But if the key is not correct, then the message will remain in the form of random characters (ciphertext).

Figure 11 shows the window when recipients choose a message to be decrypted and Figure 12 shows the result of decryption.

Figure 10. Ciphertext of message and attachment file

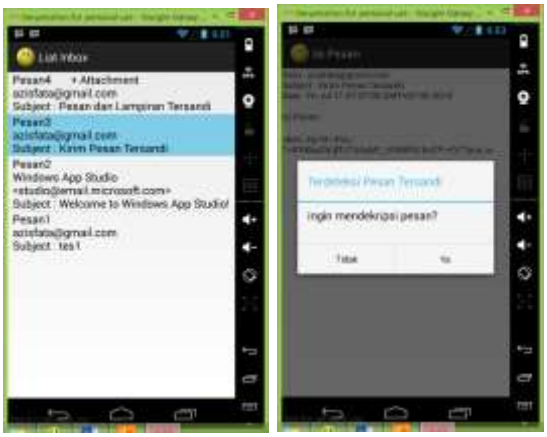

Figure 11. Recipients choose the email message

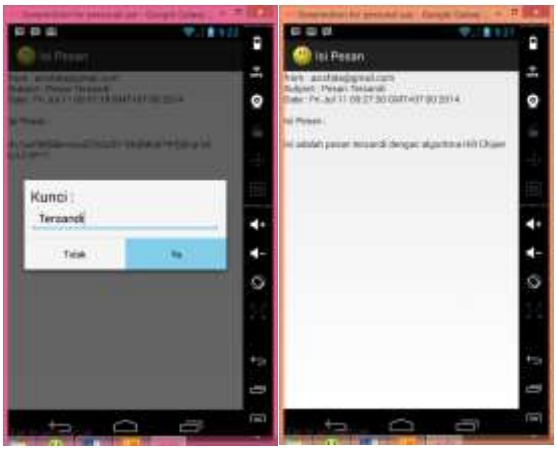

Figure 12. The result of decryption

CONCLUSSION

In order to use email security system by using a cryptography algorithm, then the two sides must install the HillMail application. Both sides referred here is the sender side and the receiver side. Based on the results, HillMail application can be used to send and receive messages properly. No one can read the contents of encrypted messages unless the legitimate recipient. Email content can be a plain message or a message with an attached file. Both the message content and attachments, are encrypted using the Hill Cipher algorithm.

Messages are secure because it is in the form of ciphertext or encrypted messages. To open an encrypted message that this required two conditions, the first is the recipient must install the HillMail application as an email client application. The second condition is the recipient must have the key to unlock the encrypted message. The key is must be the same with a key which is entered by the sender when sending a message. When users perform decryption using the wrong key, then the results are not the original email, but other forms of ciphertext.

In the cryptography process, the system runs well but the key length is limited only 9 characters. This is because the constraints in the process of decryption using Hill Cipher algorithms. To decrypt the message, the message must be multiplied by the inverse of the key used in the encryption.

Calculating the inverse matrix is different for each dimension of the matrix. Calculating the inverse matrix for the key dimension of 2x2 has a different way with a key dimension of 3x3. HillMail applications can only do the inverse of the key dimension of 2x2 and 3x3. With a dimension key of 3x3, the key can be filled with 9 characters. Thus the key can only be made up to 9 characters. If you want to use a key length of more than 9 characters then it must implement key sizes of more than 3x3.

By not considering the character flag, that is #@Qj#, the number of characters in the plaintext has an equal number to the number of characters in the ciphertext. From the results of this experiment can be concluded that the encryption algorithms using Hill Cipher produce ciphertext with the same size as the plaintext. So it does not increase the file size of the message. Email server can also receive encrypted messages such well, with no message content is missing.

REFERENCES

[1] E. Neidhardt, "Asymmetric Cryptography For Mobile Devices," *Serv.-Centric Netw.*, Pp. 1–12, 2011.

[2] P. Robinson, "Applying Cryptography As A Service To Mobile Applications," In *Rsa Conference*, 2014, Pp. 24–28.

[3] S. F. Amin And N. S. Hunnergi, "Hill Cipher Algorithm With Self Repetitive

Matrix For Secured Data Communication," In *International Journal Of Engineering Research And Technology*, 2013, Vol. 2.

[4] H. Bodur And R. Kara, "Secure Sms Encryption Using Rsa Encryption Algorithm On Android," *Message Appl. Int. Symp. Innov. Technol. Eng. Sci. Isites2015 Valencia-Spain*.

[5] A. H. Cipher, "Sistem Keamanan Attachment Email Pada Mobile Phone Android Menggunakan."

[6] T. Karlita, I. U. Nadhori, And R. I. Dewi, "Short Message Security System For Android-Based Mobile Phone Using Hill Cipher And Arithmetic Coding Algorithm," *Int. Conf. Electr. Eng. Inform. Its Educ. 2015 Ceie 2015 Univ. Malang*.

[7] C. Rahman, I. U. Nadhori, And K. Fathoni, "Studi Dan Implementasi Algoritma Blowfish Untuk Enkripsi Email," *Eepis Final Proj.*, 2010.

[8] B. Acharya, S. K. Panigrahy, S. K. Patra, And G. Panda, "Image Encryption Using Advanced Hill Cipher Algorithm," *Int. J. Recent Trends Eng.*, Vol. 1, No. 1, 2009.