Network and Complex Systems
ISSN 2224-610X (Paper) ISSN 2225-0603 (Online)
Vol.11, 2020
www.iiste.org

Data Encryption and Decryption by Using Hill Cipher Algorithm

Mebratu Fana Bedasa[1] Asrat Sime Bedada[1] Wesenu Bekele Mulatu[2]
1.Department of Computer Science, College of Computing, Madda Walabu University
2.Department of Information System, College of Computing, Madda Walabu University

Abstract
The core of Hill-cipher is matrix manipulations. It is a multi-letter cipher, for decryption the inverse of matrix requires and inverse of the matrix doesn't always exist. Then if the matrix is not invertible then encrypted text cannot be decrypted. However, a drawback of this algorithm is overcome by use of self-repetitive matrix. This matrix if 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. Hence, it derives its name i.e. self-repetitive matrix. It should be non-singular square matrix.
Key words: Hill Cipher Algorithm, Self-Repetitive Matrix and Inverse Matrix
DOI: 10.7176/NCS/11-02
Publication date: July 31st 2020

1.1 Introduction

In the recent years, authentication of information is a fundamental part of our lives as privacy. For authenticate personal or organizational data, encryption and decryption of information using different cryptographic algorithms have a key roles in wide world. Cryptography provides mechanisms for such techniques. Indeed, the protection of sensitive communications has been the emphasis of cryptography throughout much of its history [1] [2].

The Hill cipher in cryptography is used to explain the application of matrices defined over a finite field, and the handling of characters and strings in computer programs. The Hill cipher algorithm with self-repetitive matrix is one of the symmetric key algorithms that have several advantages in data encryption. But, the inverse of the key matrix used for decrypting of the cipher text does not always exist. If the key matrix is not invertible, then encrypted text cannot be decrypted. This self-repetitive Hill Cipher algorithm, initially checks the matrix used for encrypting the plaintext, whether that is invertible or not. If the encryption matrix is not invertible, then the algorithm modifies the matrix such a way that its inverse exist [3] [4] [5].

To overcome the weak security of the Hill algorithm, the proposed techniques adjusts the encryption key to form a different key for each block encryption. In the self-invertible matrix generation method, the matrix used for the encryption is itself self-invertible. Furthermore, this method eliminates the computational complexity involved in finding inverse of the matrix while decryption. So, at the time of decryption, the current study needs not to find inverse of the matrix. In order to overcome this problem, the proposed algorithm was used self-repetitive matrix. This matrix if multiplied with itself for a given mod value (i.e. mod value of the matrix is taken after every multiplication) was eventually result in an identity matrix after N multiplications. So, after N+ 1 multiplication the matrix was repeat itself. In Cipher text-only cryptanalysis of this method is very difficult.

1.2 Problem Identification

In ancient times, security of data to maintain its confidentiality, proper access control, integrity and availability has been a major issue in data communication. As soon as a sensitive message was etched on a clay tablet or written on the royal walls, then it must have been foremost in the sender's mind that the information should not get intercepted and read by a rival. When this rival get data that cannot be encrypted by using different cryptographic algorithm, data may modified or damaged by different denial of service on communication line. Data encryption and decryption by the Hill Cipher technique algorithms has several problems. The first one when data is encrypted by this method it is simple to cryptanalysis by rival since it has very weak symmetric key algorithms. The second data encrypted by this method sometimes cannot decrypt to the original plaintext. The third problem of Hill Cipher is none invertible matrices; since the encrypted text can't be decrypted. Also when the matrix not invertible, two plaintext vector will be mapped into the same cipher text vector. So, the proposed algorithms used to solve these problems is used Hill cipher with self-repetitive matrix to encrypt and decrypt data to its original plaintext [4] [5] [6] [7].

1.3 Literature Review

In this section, a few newly proposed techniques for data encryption by hill cipher have been introduced.

Bibhudendra, *et al.,* [4] proposed a novel advanced Hill (Advil) which involved an involutory matrix key in

its encryption algorithm. When an involutory key was 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. This algorithm was used to encrypt both gray scale and color images. According to this study, the proposed algorithm was more efficient when compared with the original Hill cipher. However, this algorithm did not suitable to encrypt all zeroes plaintext block.

Toorani*, et al.,* [5] they created a variant of Hill cipher which was 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. The proposed algorithm had the same structure like an affine Hill cipher. In this algorithm, each plaintext block is encrypted using a random number. This method is increasing the randomization of the algorithm, its strength towards common attacks and avoid multiple random number generation. This paper also presented a one-pass protocol for the sender and receiver to share the core random number. According to this study, the proposed algorithms were computationally efficient. But, still it had the problem of random number generation that produce a non-invertible matrix key.

Saeednia [6], he tried to make the Hill cipher secure using some random permutations of columns and rows of the key matrix but it was proved that his cryptosystem was vulnerable to the known-plaintext attack which was the same vulnerability with the original Hill cipher.

Rushdi*, et al.,* [7] proposed that the problem of non-invertible matrix key in Hill cipher. They designed a strong cryptosystem algorithm for non-invertible matrices. The non-invertible matrix key problem was solved by converting each plaintext character into two cipher text characters. So, with the decryption, the process involved the conversion of two cipher text characters into one plaintext character.

Ismail, *et al*., [8] proposed a modified Hill cipher which used a unity (one-by-one) matrix as a key to encrypt each plaintext blocks. In this paper, 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 was then multiplied with a randomly selected initial key and the multiplication results in a unique key which is used for encryption. Since the IV multiplication is performed row by row, this algorithm is known as Hill multiplying rows by initial vector (HillMRIV).

Rangel Romero*, et al*., [9] proved that the proposed algorithm was still vulnerable towards known plaintext attacks. They assumed that the key, Ki used for encryption was a 2 × 2 key matrix and the IV = [e, f] and the attacker has successfully obtained the 2 × 2 matrix key. With this key, it was possible to calculate the IV values. Apart from its vulnerability to known plaintext attack, the authors also discussed some other drawbacks of Hill cipher when all zeroes plaintext block was a matrix block where all the values in it were zero. This problem was happen when Hill cipher is used to encrypt an image which a large portions of pixels in black.

Yeh, *et al.,* [10] the proposed method used two cop-rime base numbers that were securely shared between the participants. Although this scheme thwarts the known plaintext attack, it was so time consuming, requires many mathematical manipulations, and was not efficient especially when dealing with a bulk of data.

Lin, *et al.,* [11] they tried to improve the security of the Hill cipher using several random numbers generated in a hash chain but the proposed scheme was not efficient.

1.4 Tools and Techniques

For this study, we used MATLAB (Matrix Laboratory); since it is a high-level technical computing language, interactive environment for algorithm development and used for different applications, including data visualization, data analysis, numeric computation and image processing etc. It is solving technical computing problems faster than traditional programming languages and provides all the features of programming language like arithmetic operators, flow control, data structures, data types, object-oriented programming (OOP), and debugging features.

1.5 proposed System Architecture and Algorithm

The proposed system architecture was divided into two sections.

1.5.1 Transmitter Side Process

At transmitter side, the algorithm and flowchart was designed and implemented. Then the transmitter sends block of data or files by N matrices .The key matrix is generated depends on selective matrices and data is compressed into hex code for more confusion. Data compressed into hex codes were written to file name (.txt) at sender side. The compressed hex codes data written into file name at sender sides is transmitted to receiver side through the channel.

1.5.2 Receiver side Process

At receiver side, algorithm and flowchart was designed and implemented. Then the receiver get data encoded to hex codes through the channel transmitted to it from sender side and receive the key matrix. Using this key

matrix decode data encoded into hex codes to equivalent plaintext. Finally the decoded information's were displayed at receiver side.

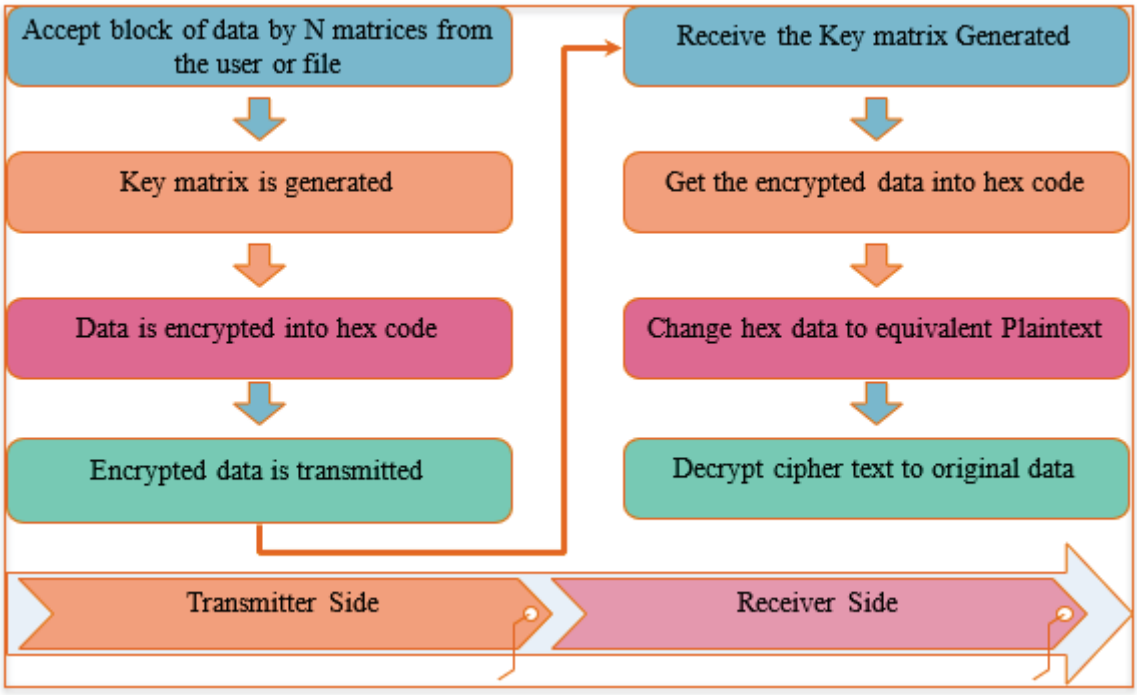

Figure 3: General Process at Transmitter and Receiver Side

1.5.3 The Hill Cipher Algorithm

This algorithm generates the different key matrix for each block encryption instead of keeping the key matrix constant. This increases the secrecy of data. Also algorithm checks the matrix used for encrypting the plaintext, whether that is invertible or not. If the encryption matrix is not invertible, then the algorithm modifies the matrix such a way that it's inverse exist. The new matrix obtained after modification of key matrix is called as encryption matrix and with the help of this matrix encryption operation is performed. In order to generate different key matrix each time, the encryption algorithm randomly generates the seed number and from this key matrix is generated [3] [4] [7].

$$\textbf{Key Matrix, K} = \begin{bmatrix} K11 & K12 & K13 \\ K21 & K22 & K23 \\ K31 & K32 & K33 \end{bmatrix}$$

where, $K11 = \text{Seed Number}$

$$k12 = (\text{seed number} * m) \bmod n$$

$$k13 = (k12 * m) \bmod n, \ldots \text{and}$$

$$k33 = (k32 * m) \bmod n$$

Where m is successive numbers of plaintext letters taken at a time for encryption and n is length of the lookup table (total characters used for encryption and decryption) or can set this n value as per requirement. Then with the help of key matrix, encryption matrix E is generated [4] [7].

1.5.4 Steps for Encryption Matrix generation

(1) Check whether the matrix K is invertible or not.

(2) If inverse of matrix K does not exist, then adjust the diagonal elements (Increment the values of diagonal elements, one element at a time) so that the inverse of the resultant matrix (matrix obtained after changing diagonal elements) is invertible. This matrix becomes the Encryption matrix E.

In this algorithm it takes m successive plaintext characters and substitutes for then m cipher text characters. The substitution is determined by m linear equations in which each character is assigned a numerical value (authors can take the character's ASCII equivalent number or can assign a lookup table like a = 0, b = 1, z = 25). Here for m = 3, the System can be described as follows [4] [7]:

$$C1 = (E11\ P1 + E12\ P2 + E13\ P3) \bmod n$$
$$C2 = (E11\ P1 + E12\ P2 + E13\ P3) \bmod n$$
$$C3 = (E11\ P1 + E12\ P2 + E13\ P3) \bmod n$$

This case can be expressed in term of column vectors and matrices:

$$\begin{pmatrix} C1 \\ C2 \\ C3 \end{pmatrix} = \begin{pmatrix} E11 & E12 & E13 \\ E21 & E22 & E23 \\ E31 & E32 & E33 \end{pmatrix} * \begin{pmatrix} P1 \\ P2 \\ P3 \end{pmatrix} \text{mod n}$$

or C = EP mod n , where C and P are column vectors of length 3 , representing the Cipher text and plaintext respectively, and E is a 3 × 3 encryption matrix. All operations are performed mod n.

1.5.5 Steps for Decryption Matrix generation

For decryption, from the seed number once again in similar way E matrix is generated. Decryption required using the modulo inverse of the matrix E. The inverse E^{-1} of matrix E is defined by the equation $E*E^{-1} = E^{-1}*E = I$ Where I is the matrix that is all zeros expect for ones along the main diagonal from upper left to lower right. Hence decryption matrix D is generated by doing modulo inverse of encryption matrix. Multiply decryption matrix D with received cipher text number vector C and then do modulo operation. Then operate on the output resultant vector, substitute its equivalent characters and which is the plaintext. This can be explained as:

Plaintext = P = D*C = E^{-1}*C. In general, the algorithm can be expressed as follows:

Cipher text = C = E*P mod n

Plain text = P = E^{-1}*C mod n = $E^{-1}E*P = P$

The flowcharts for encryption & decryption methods are represented in figures 2 & 3.

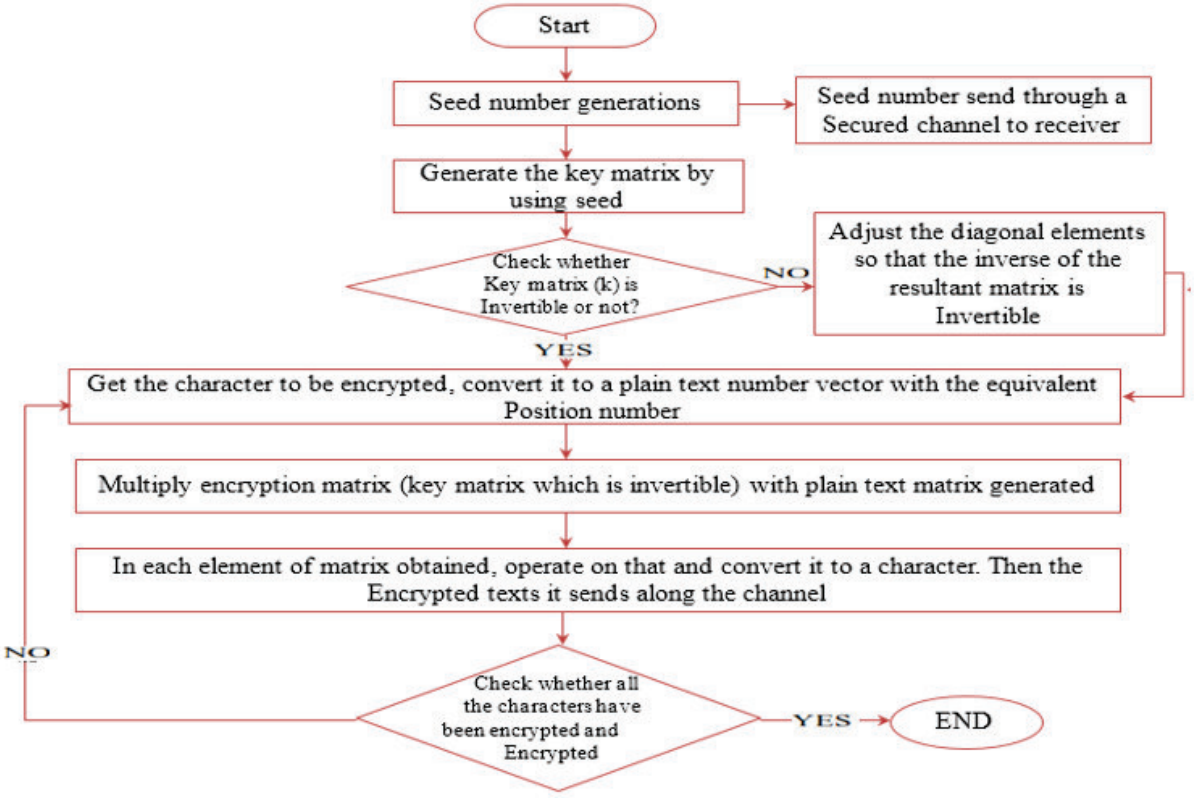

Figure 4: Flow Chart for Encryption

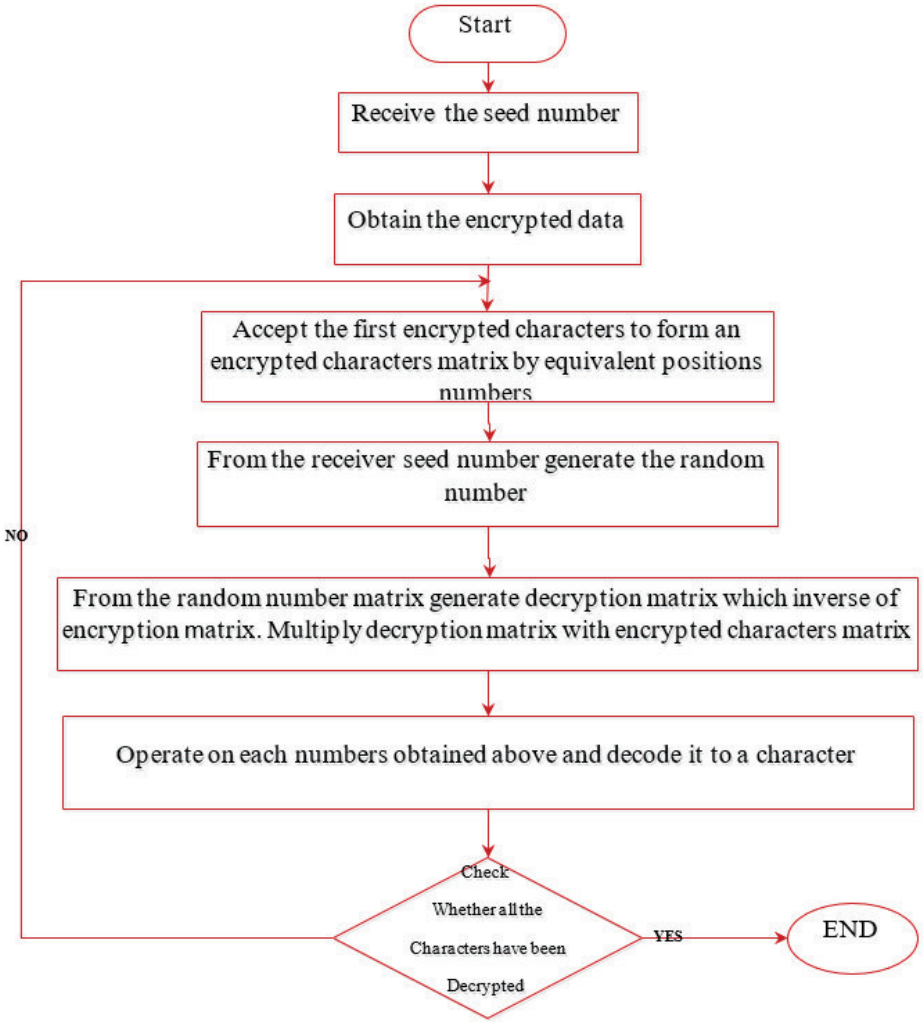

Figure 5: Flow Chart for Decryption

1.5.6 Generation of a Self-Repetitive Matrix A for a Given N:

The initial conditions for the existence of a self-repetitive matrix are:

1. The matrix should be square.
2. It should be non-singular.

But trying to find out the value of N (the value where the matrix becomes an identity matrix) through the method of brute force may not be the best idea always; because the matrix is of dimension greater than 5*5 and with mod index (i.e.) greater than 91 then the brute force technique might take very long time and N value may be in the range of millions. A normal Pentium 4 machine might hang if asked to do the computations for 15*15 matrixes or more. Hence, it would be comfortable to know the value of N and then generate a random matrix accordingly [4] [5] [7].

This can be done as follows:

1. First a diagonal matrix A is chosen, and then the values powers of each individual element when they reach unity is calculated and denoted as n1, n2, n3…. Now LCM of these values is taken to given the value of N.
2. Now the next step is generate a random square matrix whose N value is same as the N calculated in the previous step.
3. Pick up any random invertible square matrix B.
4. Generate $C=B^{-1*}A*B$
5. The N value of C is also N

The results of the following steps for encrypted data generated as follows:

Consider the plaintext to be encrypted is “event”. Letters of the plaintext are represented by their ASCII equivalent number vector (30 47 30 39 45). Then with the help of key matrix, encryption matrix is generated. Encryption matrix generated is:

$$K=\begin{matrix}36 & 0 & 0 & 0 & 0\\ 0 & 36 & 0 & 0 & 0\\ 0 & 0 & 35 & 0 & 0\\ 0 & 0 & 0 & 16 & 0\\ 0 & 0 & 0 & 0 & 35\end{matrix}$$

Then, Cipher text for the plaintext is [13 43 80 42 23].

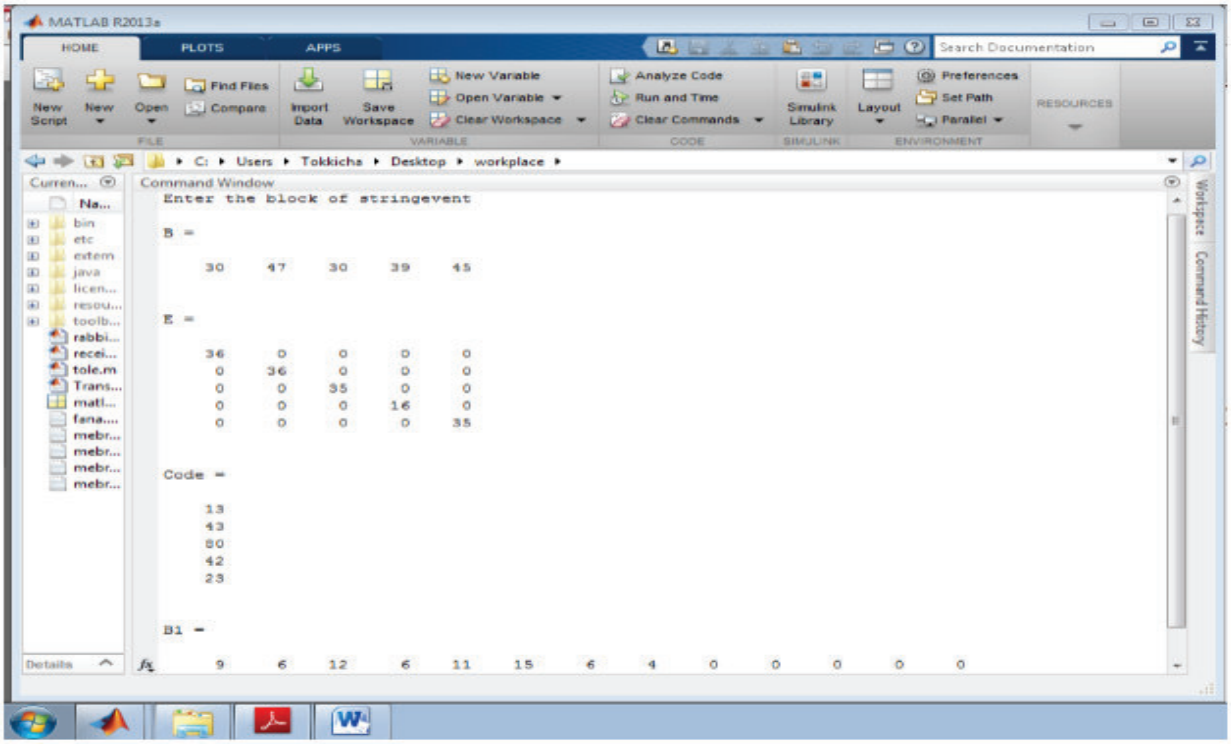

Figure 6: Sender Side Encryption

Decryption is done by doing inverse method of the above and the cipher text is converted to the original plaintext as follows:

$$\begin{matrix}30\\47\\30\\39\\45\end{matrix} = \begin{matrix}62 & 0 & 0 & 0 & 0\\ 0 & 62 & 0 & 0 & 0\\ 0 & 0 & 61 & 0 & 0\\ 0 & 0 & 0 & 91 & 0\\ 0 & 0 & 0 & 0 & 61\end{matrix} \begin{matrix}13\\43\\80\\42\\23\end{matrix} \textbf{ MOD (97)}$$

Decrypted plain text output is: Thus replacing the vector numbers (30 47 30 39 45) by their ASCII values print the word “event”.

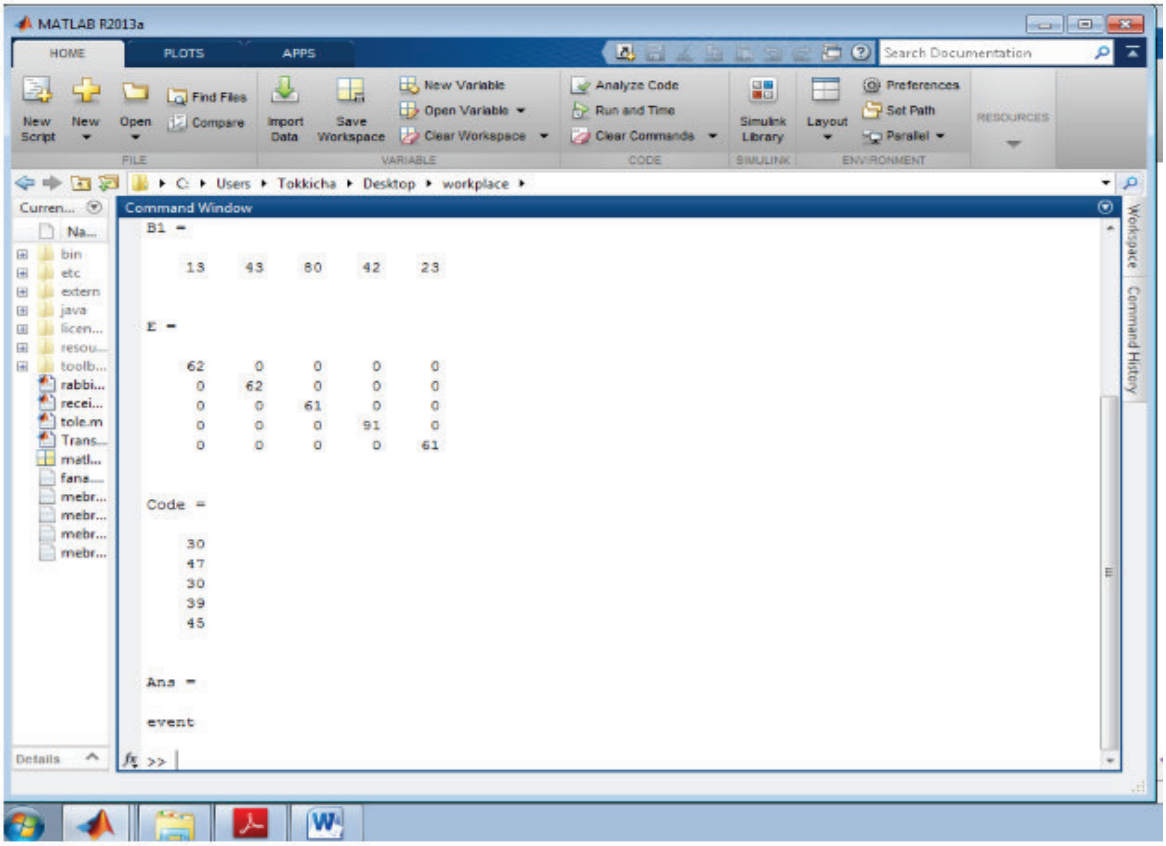

Figure 7: Receiver Side Decryption

1.5.7 General Steps of Hill Cipher Using Self-Repetitive Matrix

1.5.8 Steps at the Transmitter Side

1. Take the data in blocks of 5 as a column matrix denoted by E.
2. Multiply E the matrix generated earlier with $C^{(N-M)}$ to generate the encryption code.
3. M is some random number selected and known to both the ends. M<N.
4. Now convert this code into machine code to give better compression and bit saving.
5. Transmit (using Manchester coding or else).

1.5.9 Steps at the Receiver Side

1. Decompress the hex code into mod-97 code
2. Then multiply the code column matrix with C^{M}.
3. The corresponding substitutions are made and text recovered.

1.5.10 Method for Compression into Hex Code

1. Take the code in the place value of 97 and generate a polynomial.
2. Now divide the polynomial by 16 to generate a remainder.
3. The quotient generated forms the next dividend polynomial and division is carried out once more and remainder collected.
4. This process is carried on until divisor is larger than the dividend.
5. All the remainders are collected and the array is inverted. This is the compressed code.

Hill cipher using self-repetitive matrix has been simulated in MATLAB as follows. We considered the message for 5 blocks and simulated the algorithm on both sender and receiver side.

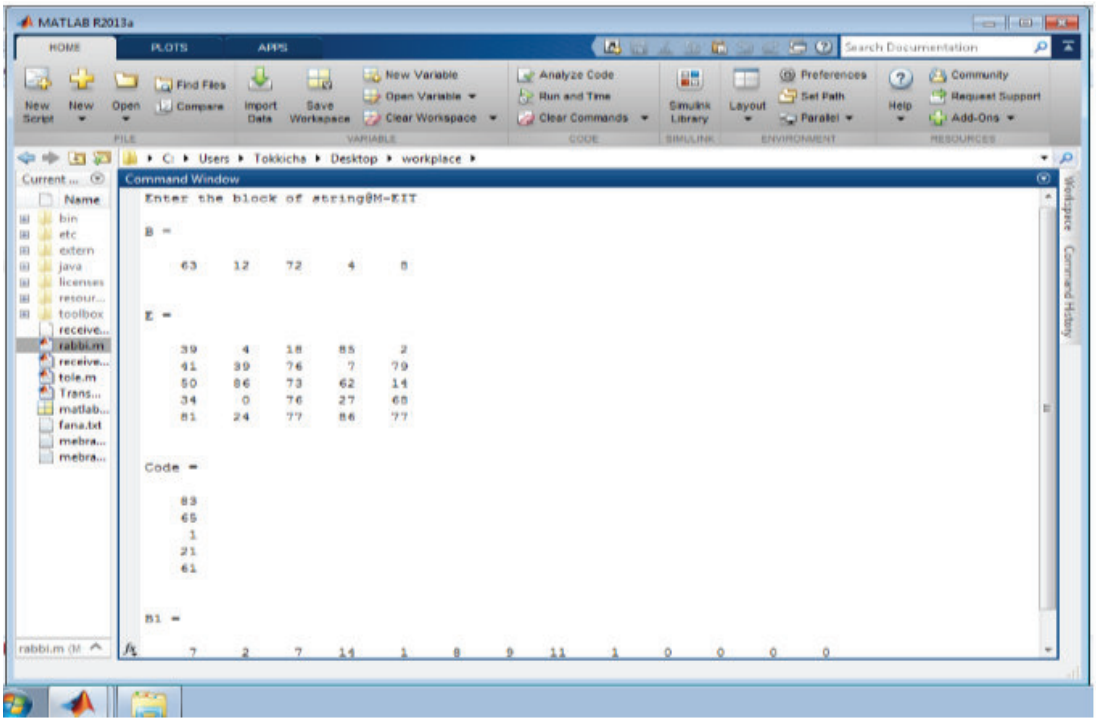

Figure 8: Transmitter Side Encryption

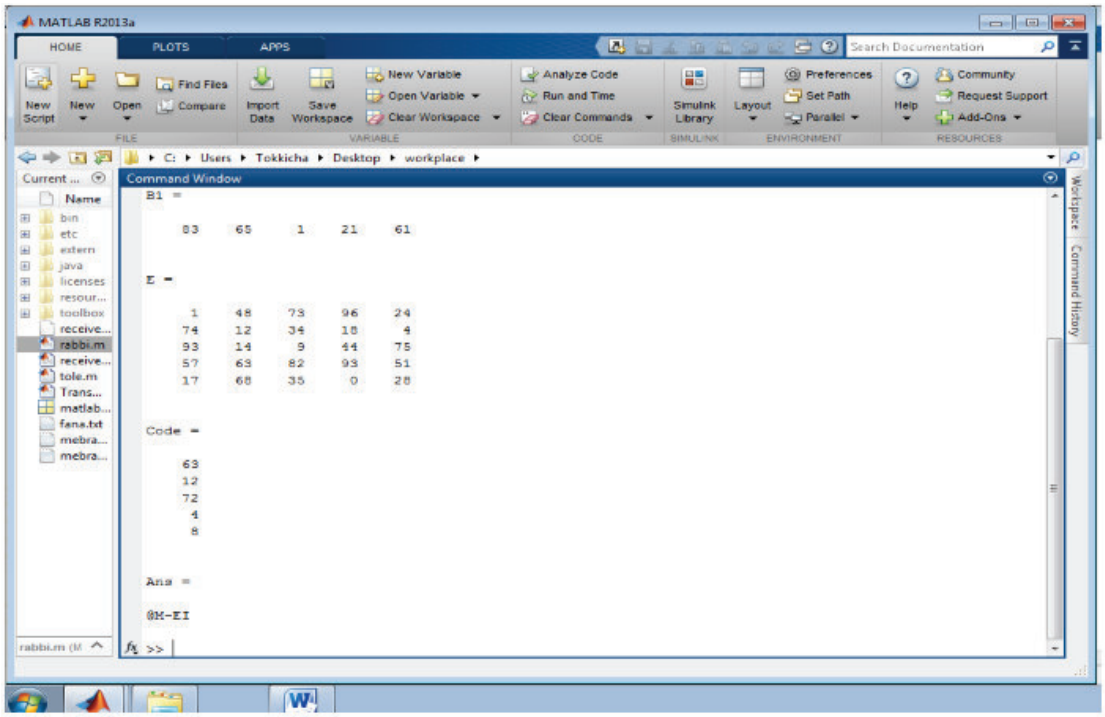

Figure 9: Receiver Side Decryption

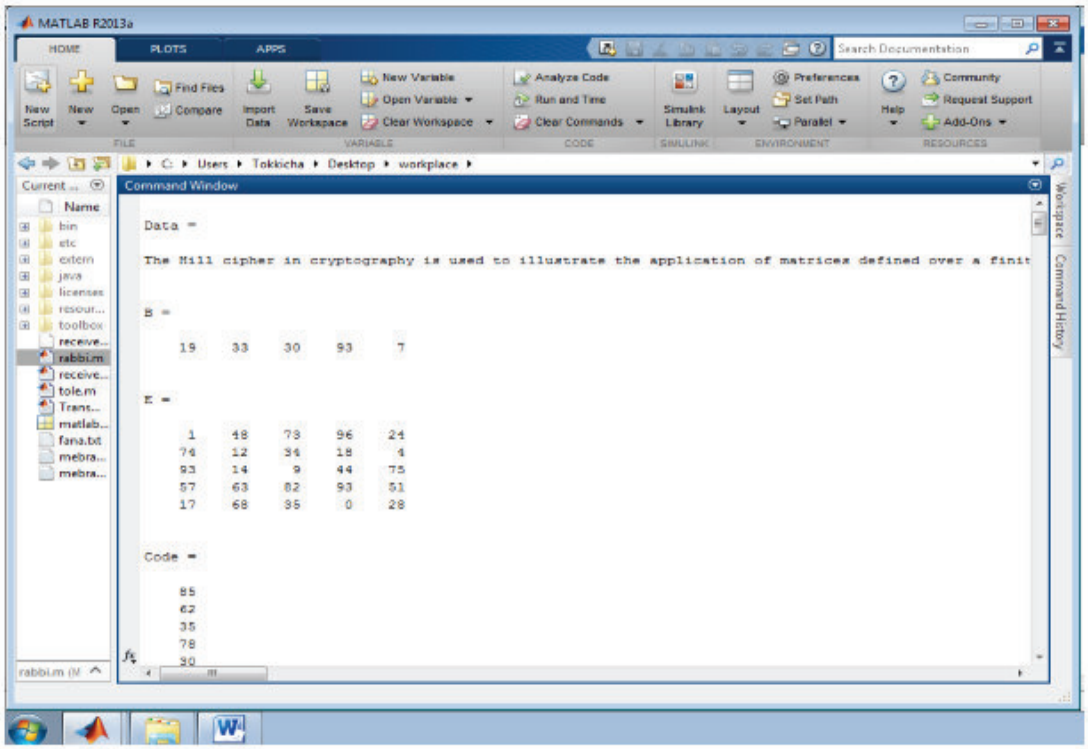

Figure 10: Transmit with Files

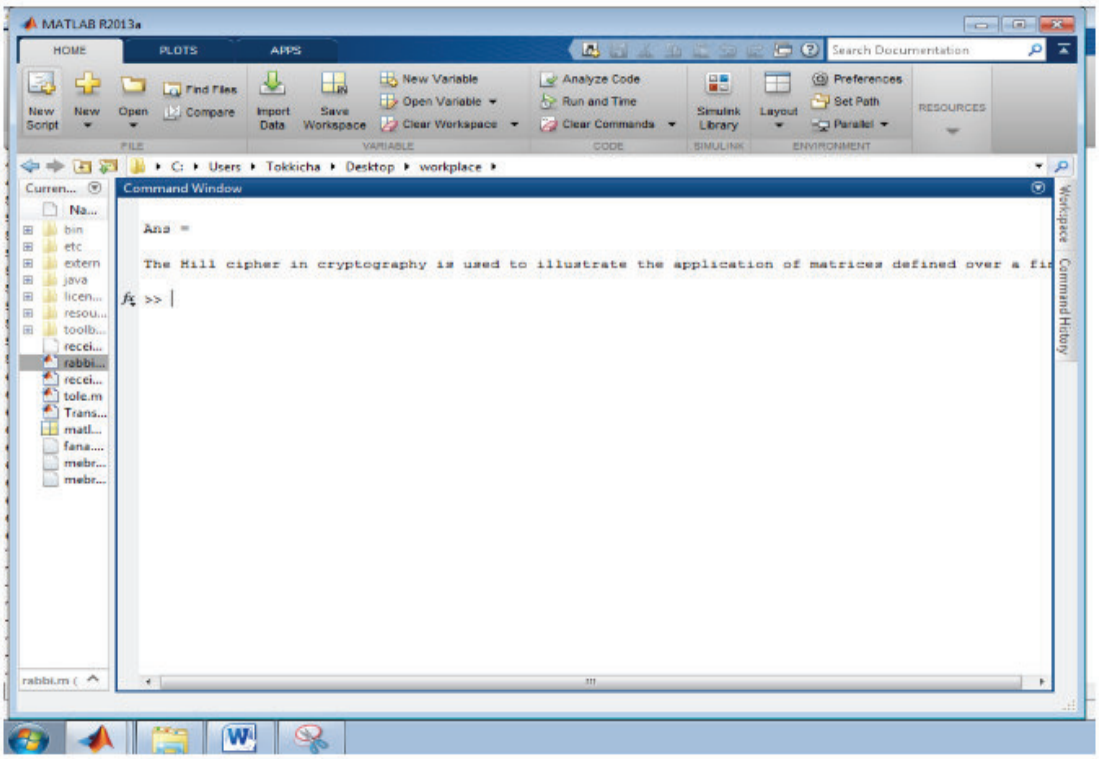

Figure 11: Receive With Files

1.5.11 Encryption and Decryption Using B (inv) AB Method

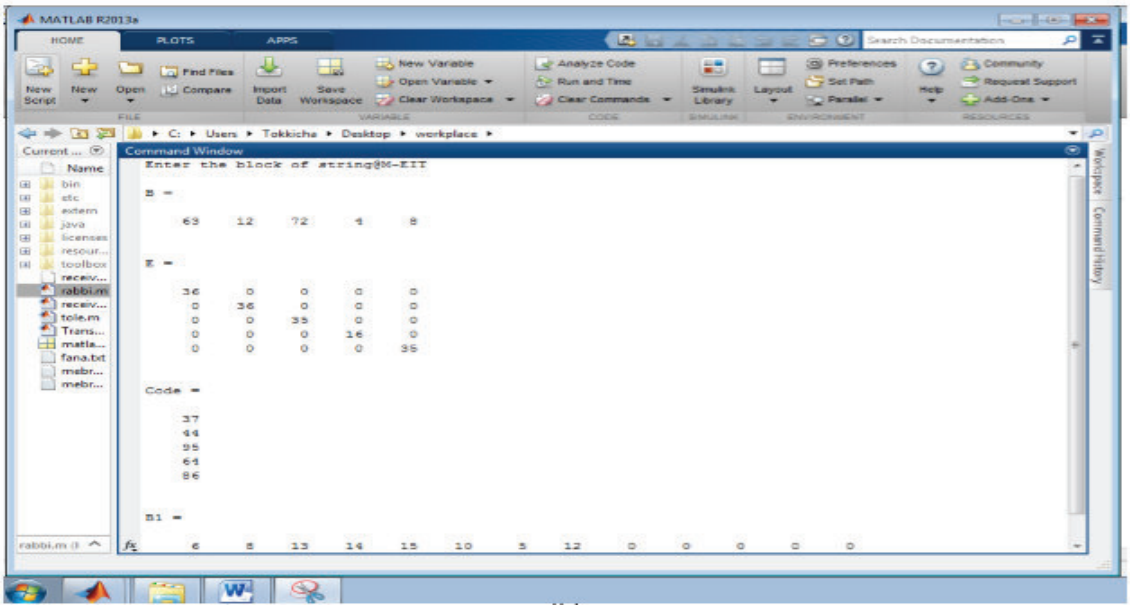

Figure 12: Transmitter Side

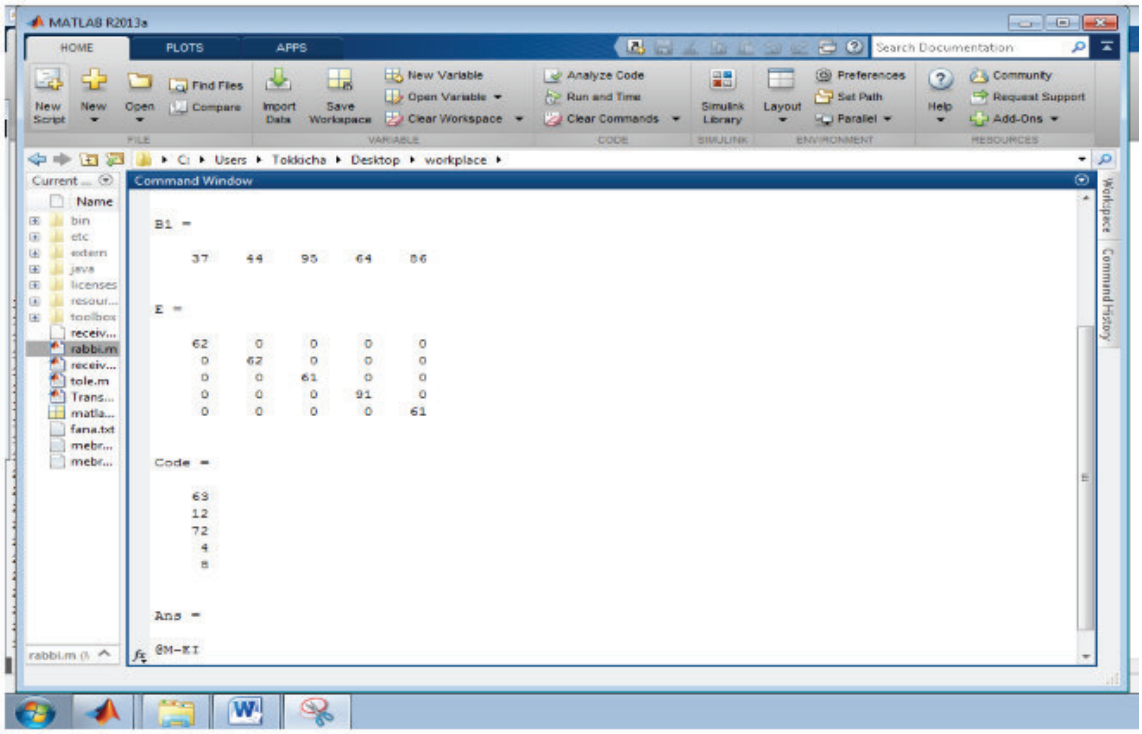

Figure 13: Receiver Side

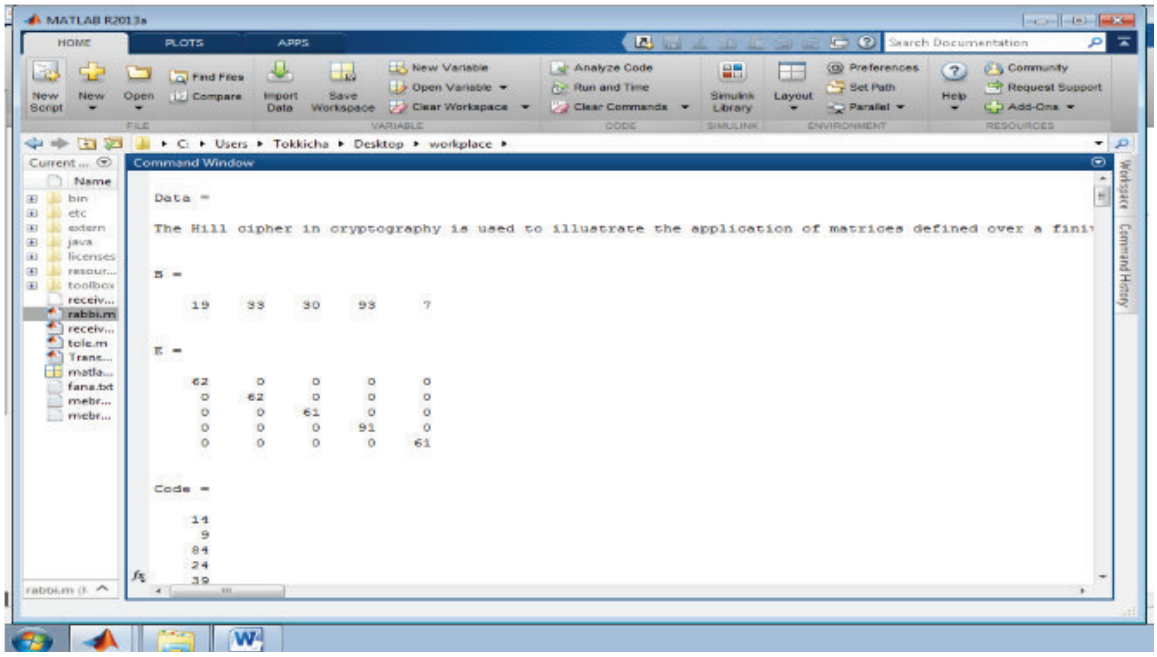

Figure 14: Transmitting With Files

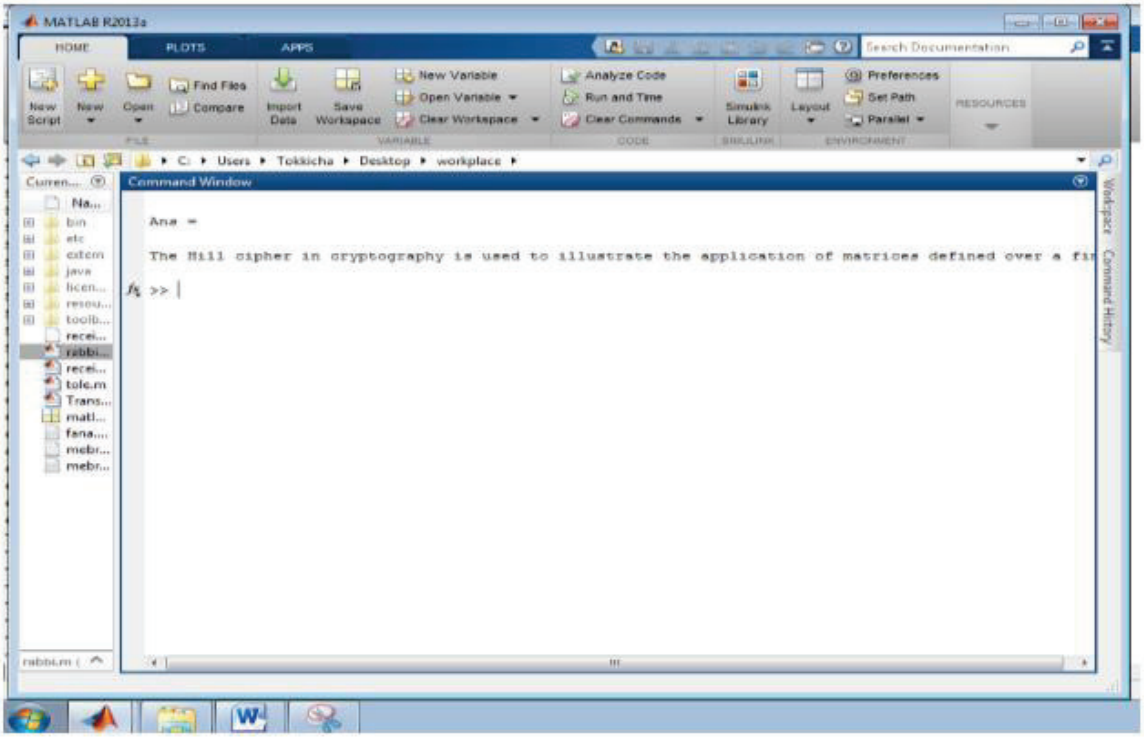

Figure 15: Receiving With Files

1.6 Conclusion and Recommendation

1.6.1 Conclusion

In general the Hill Cipher technique using a new method of self-repetitive matrix was successfully implemented. A transmitter-receiver pair was successfully modeled which used proper decompression techniques for effective communication. The numerical method suggested to find N value of a matrix was successfully tested and used in the implementation. It was found to be easier to compute and simpler to implement and difficult to crack. The above performance will be appropriate for the following kind of applications.

1) In ATMs for pin numbers to maintain its secrecy and security of ATM card.
2) In Email applications for military and civilian purpose where security is of prime importance in

terms of records and authentication of messages.

3) In SMS services, e-commerce, pay TV, computer passwords and touches many aspects of our daily lives.

1.6.2 Recommendation

Thus; the authors suggests for the future work shall be: The work has implementation for data encryption and decryption purpose. The encryption and decryption is done very well by selected cryptographic algorithm to encrypt and decrypt any alpha-numeric keys letters, numbers and symbols but it lacks to encode and decode images. Therefore, the recommended future work here is encrypted and decrypts image, music, and video using Hill Cipher with self-repetitive matrix.

1.7 Conflict of Interest

The authors would like to declare that they have no interest of conflict and we want to disclose you that it is our original research work.

1.8 References

[1] W. Stallings, Cryptography and Network Security Principles and Practice Fifth Edition, United States of America, Boston: Prentice Hall , 2007.

[2]M. S. Wiley, Information Security Principles and Practice Second Edition, San Jose, CA : A JOHN WILEY & SONS, INC., PUBLICATION , 2011.

[3]M. R. C. O. E. &. T. Department of Computer Science and Engineering, Information Security, Telangana State, India : Malla Reddy College of Engineering & Technology , 2018.

[4]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.

[5]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.

[6] S. Saeednia, "How to Make the Hill Cipher Secure," Cryptologia Journal, Vol.24, No.4, pp.353-360, Oct. 2000.

[7] A. H. Rushdi and F. Mousa, "Design of a Robust Cryptosystem Algorithm for NonInvertible Matrices Based on Hill Cipher", Int"l Journal of Computer Science and Network Security, vol. 9, no. 5, (2009), pp. 11-16.

[8] I.A. Ismail, M. Amin, and H. Diab, "How to repair the Hill cipher," Journal of Zhejiang University-Science A, Vol.7, No.12, pp.2022-2030, Dec. 2006.

[9] Y. Rangel-Romero, G. Vega-García, A. Menchaca-Méndez, D. Acoltzi-Cervantes, L. Martínez-Ramos, M. Mecate-Zambrano, F. Montalvo-Lezama, J. Barrón-Vidales, N. CortezDuarte and F. Rodríguez-Henríquez, "Comments on How to repair the Hill cipher", Journal of Zhejiang University Science A, vol. 9, no. 2, (2006), pp. 211-214.

[10] Y.S. Yeh, T.C. Wu, C.C. Chang, and W.C. Yang, "A New Cryptosystem Using Matrix Transformation," Proceedings of the 25th IEEE International Carnahan Conference on Security Technology, pp.131-138, Oct. 1991.

[11] C.H. Lin, C.Y. Lee, and C.Y. Lee, "Comments on Saeednia's improved scheme for the Hill cipher," Journal of the Chinese institute of engineers, Vol.27, No.5, pp.743-746, 2004.