MODIFICATION IN HILL CIPHER FOR CRYPTOGRAPHIC APPLICATION

Farheen Qazi
Department of Computer Engineering, Sir Syed University of Engineering and Technology Karachi. Karachi (Pakistan)
E–mail: engr.fq@gmail.com

Fozia Hanif Khan
Department of Mathematics, University of Karachi. Karachi (Pakistan)
E–mail: ms_khans2011@hotmail.com

Dur–e–Shawar Agha
Department of Computer Engineering, Sir Syed University of Engineering and Technology Karachi. Karachi (Pakistan)
E–mail: engr.dureshawaragha@gmail.com

Sadiq Ali Khan
Department of Computer Science, University of Karachi. Karachi (Pakistan)
E–mail: msakhan@uok.edu.pk

Saqib ur Rehman
Department of Mathematics, University of Karachi. Karachi (Pakistan)
E–mail: saqiburrehman@fuuast.edu.pk

Recepción: 05/03/2019 **Aceptación:** 19/03/2019 **Publicación:** 17/05/2019

Citación sugerida:
Qazi, F., Khan, F. H., Agha, D., Ali Khan, S. y ur Rehman, S. (2019). Modification in Hill Cipher for Cryptographic Application. *3C Tecnología. Glosas de innovación aplicadas a la pyme. Edición Especial, Mayo 2019*, pp. 240–257. doi: http://dx.doi.org/10.17993/3ctecno.2019.specialissue2.240–257

Suggested citation:
Qazi, F., Khan, F. H., Agha, D., Ali Khan, S. & ur Rehman, S. (2019). Modification in Hill Cipher for Cryptographic Application. *3C Tecnología. Glosas de innovación aplicadas a la pyme. Special Issue, May 2019*, pp. 240–257. doi: http://dx.doi.org/10.17993/3ctecno.2019.specialissue2.240–257

ABSTRACT

In order to keep the information secure from various contenders is an important factor for data security. For any organization, it is an incredibly important feature to timely transmit secured data. Optimized techniques for key management and protected encryption algorithms are always helpful for reducing the overhead of the system and maintain the integrity, authentication and confidentiality of data. Cryptographic applications play an important role in our daily lives through sending emails, exchanging bank account transaction information, through mobile communication and through ATM card transaction. To secure our information from unauthorized users, Hill Cipher is one of the most well-known symmetric cryptosystems. For Hill Cipher, here we are proposed an algorithm for encryption and decryption which is based upon the transposition, substitution and left-right shift.

KEYWORDS

Traditional Hill Cipher (THC), Transposition Substitution (TS), Transposition Substitution & Left Right Shift (TSLRS), Cryptography, Encryption, Decryption.

DOI: http://dx.doi.org/10.17993/3ctecno.2019.specialissue2.240-257

1. INTRODUCTION

In today's era of information technology, security is highly essential for our sensitive or important data. Our primary concern is to transmit data or information in a secured manner. Cryptography not only protects our information from intruders but also maintains the data integrity, confidentiality and user authenticity. We have proposed a new and secured procedure in the paper (Khan, Shams, Qazi & Agha, 2015) for generating orthogonal matrix which uses as a key matrix in Hill Cipher for enhancing the security, as well as minimize the cost of time in decryption procedure. In another paper (Khan & Qazi, 2018) we have presented a procedure of encryption and decryption to improve the security of Hill Cipher by adopting transposition and substitution technique.

In the current paper, we have to use transposition and substitution technique as defined in (Khan & Qazi, 2018). To reduce the computational complexity orthogonal key matrix is used as described in (Khan, *et al.*, 2015). An advanced procedure for encryption and decryption is proposed for making Hill Cipher more secure and efficient. Hill cipher can be more secure by applying transposition, substitution and bit shifting on original message for encryption because Hill cipher is vulnerable to a known-plaintext attack and it is completely linear but here we combined it with other non-linear operations (transposition, substitution and left-right bit shifting) which provide good diffusion. In this methodology TSLRS techniques have applied on Hill Cipher due to this, efficient outputs are obtained.

2. LITERATURE REVIEW

Saeednia (2000) presents modification on Hill Cipher and proposed a symmetric cipher. Generate a key for encryption of message by applying random permutation on rows and columns. Mohsen and Abolfazl (2011) presented an encryption algorithm to improved the modification of the Affin Hill Cipher and have introduced two more protocols. Through affine transformation, an efficient cryptosystem has created. Kim and Lee (2004) have worked on private and

public key crypto processor regarding its design and implementation. The main goal for creating this cryptographic algorithm is to optimize the execution of the microprocessor. Various security applications can apply this crypto processor like: network router security, storage devices and embedded systems. Sukhraliya, Chaudhary and Solanki (2013) presented an algorithm in which numbers are randomly generated and calculate the modulus and remainder of the numbers. Due to this new method for encrypting and decrypting, three or more keys are generated which make the ciphering technique more complicated. Acharya, Rath, Patra and Panigrahy (2007) have proposed a method of generating self-invertible matrix for Hill Cipher. For encrypting the plaintext we need the inverse of the matrix. Decryption cannot be performed on data if the matrix is not invertible.

Magamba, Kadaleka and Kasambara (2012) worked on Hill Cipher. According to the technique, the plaintext is broken into blocks of m size and multiply it by m*m matrix obtaining variable-key length matrix from Maximum Distance Separable (MDS) master key matrix. Sastry and Janaki (2008) developed a block cipher technique by modifying the Hill Cipher. They have mixed binary bits of key matrix and plaintext at a different level of iteration. Krishna and Madhuravani (2012) proposed a randomized approach for Hill Cipher; broken the plaintext into equal block size and encrypt the block one by one. The output is randomized for one plaintext that is able to generate multiple ciphertext. Sastry, *et al.* (2009) proposed an iterative method of modification in Hill Cipher in three stages. Different functions are used in these stages like inverse, matrix mixing and XOR. Hamamreh and Farajallah (2009) have presented a new model of Hill Cipher by using quadratic residue. Sastry, *et al.* (2011) proposed the new idea of permuted key and presented as a generalized advanced hill cipher. In the iterations, we find the arithmetic and mix column operation which entailed in the cipher. Binary bits of the key mix with the plaintext in a detailed manner and due to this avalanche effect and cryptanalysis makes the ciphertext more strong. Varanasi, *et al.* (2011) presented a symmetric block cipher which contains a pair of keys, iteration process, modular arithmetic addition, substitution and mixing. To strengthen the cipher significantly, they have mixed the bits and substitution in each round of iteration.

DOI: http://dx.doi.org/10.17993/3ctecno.2019.specialissue2.240-257

Keliher and Delaney (2013) worked on two variants of the classical hill cipher introduced by Toorani and Falahati. They have designed a system having an ability to overcome the weaknesses of the hill cipher and are resistant to any of the attacks i-e. ciphertext attack, known-plaintext attack, chosen-plaintext attack or chosen-ciphertext attack. Toorani-Falahati hill cipher can easily break through the described chosen-plaintext attack and confirms the effectiveness of the presented attack. Levine and Chandler (1989) proposed an algorithm of cryptographic equations relating ciphertext, plaintext and if the cipher letters are unknown the elements of the key matrix of hill system have a non-linear system of equations. To reduce these equations to the linear system they had introduced a large set of the unknown, if the plaintext is known. Sharma and Rehan (2013) proposed two-fold securities to the existing hill cipher by using logical operations and elements of the finite field.

3. PROPOSED METHODOLOGY

In this procedure, we combine previously proposed traditional hill cipher procedures with additional transposition, substitution and left-right shifting functions.

In this method, transposition and substitution of plaintext are performed on n*n matrix respectively. After this second procedure of right and left shifting is applied on the encryption process, all encryption functions are applied in reverse order to perform the decryption process.

Previously we have described the complete procedure of key generation (Khan, *et al.*, 2015). The generated orthogonal key is,

$$k = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \\ k_{41} & k_{42} & k_{43} & k_{44} \end{bmatrix}$$

We have some negative values in the generated key so, by taking additive inverse in mod 26 make all the values positive and for further simplification, we will take mod 26, due to this secured key will be generated, after that $\boldsymbol{k}$ will turn into $\boldsymbol{k'}$:

$$k'=\begin{bmatrix} k'_{11} & k'_{12} & k'_{13} & k'_{14} \\ k'_{21} & k'_{22} & k'_{23} & k'_{24} \\ k'_{31} & k'_{32} & k'_{33} & k'_{34} \\ k'_{41} & k'_{42} & k'_{43} & k'_{44} \end{bmatrix} \quad (1)$$

3.1. PROPOSED METHOD FOR ENCRYPTION

Previously we have introduced a new method for securing the hill cipher algorithm. To make hill cipher algorithm more secure and more powerful we can combine both methods consider any plaintext :

$$p=\begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}$$

The first step is to apply transposition on plaintext in reverse order so, plaintext becomes:

$$p'=\begin{bmatrix} p_4 \\ p_3 \\ p_2 \\ p_1 \end{bmatrix}$$

In this way generate a new key by using the Caesar cipher substitution on $\boldsymbol{p'}$, which is obtained by picking the values,

DOI: http://dx.doi.org/10.17993/3ctecno.2019.specialissue2.240-257

$$p"=\begin{bmatrix} p"_4 \\ p"_3 \\ p"_2 \\ p"_1 \end{bmatrix}$$

convert the above plaintext into binary form then perform right shifting so, will be similar to:

$$p'''=\begin{bmatrix} b_4 \\ b_3 \\ b_2 \\ b_1 \end{bmatrix}$$

Convert into decimal number so, is newly produces plaintext:

$$p""=\begin{bmatrix} b'_4 \\ b'_3 \\ b'_2 \\ b'_1 \end{bmatrix}$$

Now, applying the general method of hill cipher for encryption so will be similar to:

$$\begin{bmatrix} k'_{11} & k'_{12} & k'_{13} & k'_{14} \\ k'_{21} & k'_{22} & k'_{23} & k'_{24} \\ k'_{31} & k'_{32} & k'_{33} & k'_{34} \\ k'_{41} & k'_{42} & k'_{43} & k'_{44} \end{bmatrix} * \begin{bmatrix} b'_4 \\ b'_3 \\ b'_2 \\ b'_1 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} \quad (2)$$

Where **c** is the newly generated ciphertext**.**

3.2. PROPOSED METHOD FOR DECRYPTION

As we discuss in the beginning we are using orthogonal key, therefore, the inverse of the orthogonal key will be the transpose of :

$$k' = k'^{t} = k'^{-1}$$

$$k'^{-1} * k'^{t} = I$$

Later than, apply the general method of decryption for hill cipher so, will be produced

$$\begin{bmatrix} k'_{11} & k'_{12} & k'_{13} & k'_{14} \\ k'_{21} & k'_{22} & k'_{23} & k'_{24} \\ k'_{31} & k'_{32} & k'_{33} & k'_{34} \\ k'_{41} & k'_{42} & k'_{43} & k'_{44} \end{bmatrix} * \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} b'_4 \\ b'_3 \\ b'_2 \\ b'_1 \end{bmatrix} \quad (3)$$

Now, convert into binary form and perform left shifting on it

$$p''' = \begin{bmatrix} b_4 \\ b_3 \\ b_2 \\ b_1 \end{bmatrix}$$

Apply back substitution of poly alphabetic technique on so, it will similar to:

$$p'' = \begin{bmatrix} p''_4 \\ p''_3 \\ p''_2 \\ p''_1 \end{bmatrix}$$

Apply transposition in reverse order and get the original text

$$\begin{bmatrix} p_4 \\ p_3 \\ p_2 \\ p_1 \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}$$

4. ALGORITHM

i. Step 1: Take any plain text.

ii. Step 2: Apply the simple transposition on the plane text as defined in section.

DOI: http://dx.doi.org/10.17993/3ctecno.2019.specialissue2.240-257

iii. Step 3: Apply the Caesar cipher substitution as defined in section

iv. Step 4: Apply the right shift procedure after step 3.

v. Step 5: Apply the procedure of encryption defined by section 3.1.

vi. Step 6: Perform the procedure of decryption explained in section 3.2.

5. EXAMPLE

In this procedure, the plaintext "SECURITY" is divided into a block size of four letters with the key size of 4*4. For encryption process, we can apply simple transposition, Caesar cipher substitution and one-bit right shift operation on the plaintext. After this apply the regular procedure of Hill cipher and obtain the ciphertext. For decryption process, multiply the inverse of the key with the ciphertext then apply one bit left shift operation, reverse substitution and reverse transposition so, we get the original plaintext "SECU".

Consider an orthogonal key which we have generated through the previous defined procedure.

$$K=\begin{bmatrix}77 & -14 & -42 & -56\\ -14 & 98 & -21 & -28\\ -42 & -21 & 42 & -84\\ -56 & -28 & -84 & -7\end{bmatrix}$$

After taking additive inverse and mod 26, ***k*** will become ***k'***:

$$k'=\begin{bmatrix}25 & 12 & 10 & 22\\ 12 & 20 & 05 & 24\\ 10 & 05 & 16 & 20\\ 22 & 24 & 20 & 19\end{bmatrix}$$

Encryption

Consider any plaintext :

plaintext = SECURITY

We are taking four (4) letters of the plaintext at a time:

$$p = \begin{bmatrix} S \\ E \\ C \\ U \end{bmatrix} = \begin{bmatrix} 18 \\ 4 \\ 2 \\ 20 \end{bmatrix}$$

Applying transposition

$$p' = \begin{bmatrix} U \\ C \\ E \\ S \end{bmatrix} = \begin{bmatrix} 20 \\ 2 \\ 4 \\ 18 \end{bmatrix}$$

Applying Caesar Cipher Substitution

$$p'' = \begin{bmatrix} 20+7 \\ 2+7 \\ 4+7 \\ 18+7 \end{bmatrix} = \begin{bmatrix} 27 \\ 9 \\ 11 \\ 25 \end{bmatrix} (\text{mod } 26) = \begin{bmatrix} 1 \\ 9 \\ 11 \\ 25 \end{bmatrix}$$

Right Shifting

$$p''' = \begin{bmatrix} 1 \\ 9 \\ 11 \\ 25 \end{bmatrix} = \begin{bmatrix} 00000001 \\ 00001001 \\ 00001011 \\ 00011001 \end{bmatrix} = \begin{bmatrix} 00000000 \\ 00000100 \\ 00000101 \\ 00001100 \end{bmatrix}$$

Save the shifted bits in a variable **shft**

$$\text{Shft} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Binary to decimal conversion

DOI: http://dx.doi.org/10.17993/3ctecno.2019.specialissue2.240-257

$$p''''= \begin{bmatrix} 0 \\ 4 \\ 5 \\ 12 \end{bmatrix}$$

For encryption follow the normal method of hill cipher, so, will be as follow:

$$\begin{bmatrix} 25 & 12 & 10 & 22 \\ 12 & 20 & 05 & 24 \\ 10 & 05 & 16 & 20 \\ 22 & 24 & 20 & 19 \end{bmatrix} * \begin{bmatrix} 0 \\ 4 \\ 5 \\ 12 \end{bmatrix} = \begin{bmatrix} 362 \\ 393 \\ 340 \\ 424 \end{bmatrix} (\text{mod } 26)$$

$$c = \begin{bmatrix} 24 \\ 3 \\ 2 \\ 8 \end{bmatrix}$$

Decryption

After the regular procedure of decryption of hill cipher we produce the plaintext:

$$\begin{bmatrix} 25 & 12 & 10 & 22 \\ 12 & 20 & 05 & 24 \\ 10 & 05 & 16 & 20 \\ 22 & 24 & 20 & 19 \end{bmatrix} * \begin{bmatrix} 24 \\ 3 \\ 2 \\ 8 \end{bmatrix} = \begin{bmatrix} 832 \\ 550 \\ 447 \\ 792 \end{bmatrix} (\text{mod } 26)$$

$$p'''' = \begin{bmatrix} 0 \\ 4 \\ 5 \\ 12 \end{bmatrix}$$

Left Shifting

$$p''' = \begin{bmatrix} 0 \\ 4 \\ 5 \\ 12 \end{bmatrix} = \begin{bmatrix} 00000000 \\ 00000100 \\ 00000101 \\ 00001100 \end{bmatrix} = \begin{bmatrix} 00000000 \\ 00001000 \\ 00001010 \\ 00011000 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 00000001 \\ 00001001 \\ 00001011 \\ 00011001 \end{bmatrix}$$

Binary to decimal conversion

$$p'' = \begin{bmatrix} 1 \\ 9 \\ 11 \\ 25 \end{bmatrix}$$

Perform back substitution

$$p'' = \begin{bmatrix} 1-7 \\ 9-7 \\ 11-7 \\ 25-7 \end{bmatrix} = \begin{bmatrix} -6 \\ 2 \\ 4 \\ 18 \end{bmatrix} (\text{mod } 26) = \begin{bmatrix} 20 \\ 2 \\ 4 \\ 18 \end{bmatrix}$$

Perform back transposition and get original plaintext

$$\begin{bmatrix} 20 \\ 2 \\ 4 \\ 18 \end{bmatrix} = \begin{bmatrix} 18 \\ 4 \\ 2 \\ 20 \end{bmatrix} = \begin{bmatrix} S \\ E \\ C \\ U \end{bmatrix}$$

In the way, the plain text will be recover.

6. EXPERIMENTAL RESULTS

To quantify the experimental results different parameters are used to obtain the execution of this algorithm and their comparison with the existing method. Through simulation, experimental results are obtained by using MATLAB. Table 1 shows the overall performance evaluation and encryption time, decryption time and the orthogonal key evaluation time.

Table 1. Encryption & Decryption execution time of Actual and Proposed algorithms.

File Size (bytes)	Encryption & Decryption Execution Time (msec)	
	THC	TSLRS
10	67	65
14	109	106
18	132	116
Total: 42	308	287

DOI: http://dx.doi.org/10.17993/3ctecno.2019.specialissue2.240-257

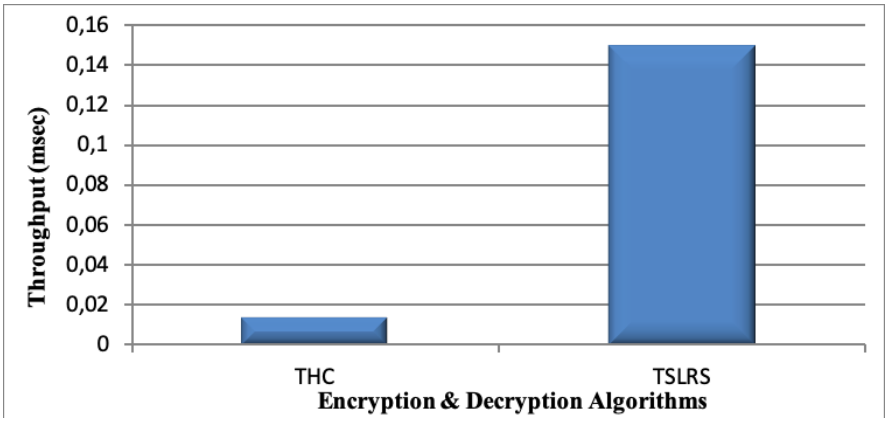

Figure 1. Encryption & Decryption execution time comparison of the actual algorithm with the proposed algorithms using the different file size.

Calculation of Encryption & Decryption Throughput:

Decryption Throughput (bytes/sec) = Σ Input File Size/Σ Encryption & Decryption Execution Time

Σ Input file Size = 10 + 14 + 18

Σ Input file Size = 42 bytes.

Encryption & Decryption Throughput for THC:

Σ Encryption & Decryption Execution Time [THC] = 67+109+132

Σ Encryption & Decryption Execution Time [THC] = 308

Encryption & Decryption Throughput [THC] =42/308

Encryption & Decryption Throughput [THC] = 0.014 bytes/msec.

Encryption & Decryption Throughput for TSLRS:

Σ Encryption & Decryption Execution Time [TSLRS] = 65+106+116

Σ Encryption & Decryption Execution Time [TSLRS] = 287

Encryption & Decryption Throughput [TSLRS] =42/287

Encryption & Decryption Throughput [TSLRS] = 0.15 bytes/msec.

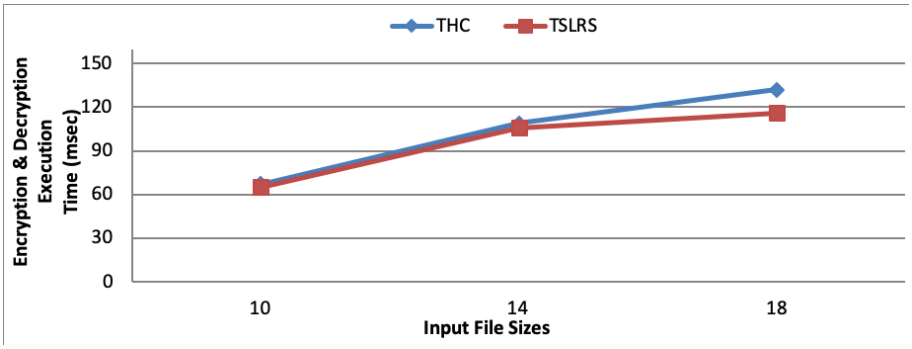

Figure 2. Throughput of THC & TSLRS Encryption & Decryption Algorithms.

7. CONCLUSION

From the above experimental results, it has examined that the proposed algorithm provides optimized results in comparison with the inverse process as well as the encryption and decryption algorithms. This algorithm provides better throughput for file sizes of any type when compared with the actual algorithm.

DOI: http://dx.doi.org/10.17993/3ctecno.2019.specialissue2.240-257

REFERENCES

Acharya, B., Rath, G. S., Patra, S. K. & Panigrahy, S. K. (2007). Novel methods of generating self-invertible matrix for hill cipher algorithm. *International Journal of Security, 1*(1), pp. 14-21.

Hamamreh, R. & Farajallah, M. (2009). Design of a robust cryptosystem algorithm for non-invertible matrices based on hill cipher. *International Journal of Computer Science and Network Security, 9*, pp. 11-16.

Keliher, L. & Delaney, A. Z. (2013). Cryptanalysis of the toorani-falahati hill ciphers. In 2013 IEEE Symposium on Computers and Communications (ISCC) pp. 436-440. IEEE.

Khan, F. H. & Qazi, F. (2018). Advance Procedure Of Encryption And Decryption Using Transposition And Substitution. *Journal of Information Communication Technologies and Robotic Applications*, pp. 43-56.

Khan, F. H., Shams, R., Qazi, F. & Agha, D. (2015). Hill Cipher Key Generation Algorithm by using Orthogonal Matrix. In Proceedings International Journal of Innovative Science and Modern Engineering, 3(3), pp. 5-7.

Kim, H. W. & Lee, S. (2004). Design and implementation of a private and public key crypto processor and its application to a security system. IEEE Transactions on Consumer Electronics, 50(1), pp. 214-224.

Krishna, A. V. N. & Madhuravani, K. (2012). A modified Hill cipher using randomized approach. *International Journal of Computer Network and Information Security, 4*(5), pp. 56-62. doi: http://dx.doi.org/10.5815/ijcnis.2012.05.07

Levine, J. & Chandler, R. (1989). The Hill cryptographic system with unknown cipher alphabet but known plaintext. *Cryptologia, 13*(1), pp. 1-28. doi: http://dx.doi.org/10.1080/0161-118991863736

Magamba, K., Kadaleka, S. & Kasambara, A. (2012). Variable-length Hill Cipher with MDS Key Matrix. arXiv preprint arXiv: https://arxiv.org/abs/1210.1940

Mohsen, T. & Abolfazl, F. (2011). A Secure Cryptosystem based on Affine Transformation. *Journal of Security and Communication Networks, 4*(2), pp. 207-215. doi: http://dx.doi.org/10.1002/sec.137

Saeednia, S. (2000). How to Make the Hill Cipher Secure. *Cryptologia, 24*(4), pp. 353-360. doi: http://dx.doi.org/10.1080/01611190008984253

Sastry, V. U. K. & Janaki, V. (2008). A modified hill cipher with multiple keys. *International Journal of Computational Science, 2(*6), pp. 815-826.

Sastry, V. U. K., Murthy, D. S. R. & Bhavani, S. D. (2009). A Block Cipher Involving a Key Applied on Both the Sides of the Plain Text. *International Journal of Computer and Network Security (IJCNS), 1*(1), pp. 27-30.

Sastry, V. U. K., Varanasi, A. & Kumar, S. U. (2011). A Modern Advanced Hill Cipher Involving a Permuted Key and Modular Arithmetic Addition Operation. *Journal of Global Research in Computer Science, 2*(4), pp. 92-97.

Sharma, P. L. & Rehan, M. (2013). On Security of Hill Cipher using Finite Fields. *International Journal of Computer Applications, 71*(4), pp. 30-33. doi: http://dx.doi.org/10.5120/12348-8637

Sukhraliya, V., Chaudhary, S. & Solanki, S. (2013). Encryption and Decryption Algorithm using ASCII values with substitution array Approach. *International Journal of Advanced Research in Computer and Communication Engineering, 2(*8), pp. 3094-3097.

Varanasi, A., Sastry, V. U. K. & Kumar, S. U. (2011). A modern Advanced Hill cipher Involving a Pair of Keys, Modular Arithmetic Addition and Substitution. *Journal of Global Research in Computer Science, 2*(5), pp. 58-65.

Edición Especial Special Issue Mayo 2019
DOI: http://dx.doi.org/10.17993/3ctecno.2019.specialissue2.240-257