www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issue 10 Oct. 2016, Page No. 18265-18269

An Efficient Approach for Secure Data Hiding Using Cryptography

Reetu[1], Ms.Mamta[2], Mr. Satish Dhull[3]

[1]Department of Computer Science
South Point Institute of Technology and Management(SITM),
Deenbandhu Chhotu Ram University of Science & Technology (DCRUST), Sonepat
reetu.bhullar5@gmail.com
[2]Department of Computer Science
South Point Institute of Technology and Management(SITM),
Deenbandhu Chhotu Ram University of Science & Technology (DCRUST), Sonepat
Mamtakalra21@gmail.com
[3]satishdhull.708@gmail.com

Abstract— Cryptography is a technique used to avoid unauthorized access of data. It has two main components; a) Encryption algorithm, and b) Key. Sometime, multiple keys can also be used for encryption. A number of cryptographic algorithms are available in market such as DES, AES, TDES and RSA. The strength of these encryption algorithms depends upon their key strength. Strong encryption algorithms and optimized key management techniques always help in achieving confidentiality, authentication and integrity of data and reduce the overheads of the system. The long key length takes more computing time to crack the code and it becomes difficult for the hacker to detect the cryptographic model. In this paper we suggest an innovation in the age-old conventional cryptographic technique of HILL-CIPHER using the concept of self repetitive matrix. A numerical method has been stated, mathematically proved and later implemented in generating a random matrix of given periodicity. The method of self-repetitive matrix has then been used to simulate a communication channel with proper decompression techniques to facilitate bit saving.

Keywords— Symmetric cryptography, Asymmetric cryptography, Hill Cipher, self-repetitive matrix

1 Introduction

In today's world cryptography has become a necessity for all the organizations. Data security is an essential component of an organization in order to keep the information safe from various competitors. It also helps to ensure the privacy of a user from others. These days' passwords are not considered as reliable for this task because it is easy to guess passwords due to its short range. Moreover, if the range of password is small a brute force search can be applied to crack it [3]. So, as to protect our data various algorithms have been designed. It helps us to securely access bank accounts, electronic transfer of funds and many more daily life applications.

Cryptography [1] is a technique used to avoid unauthorized access of data. It has two main components; a) Encryption algorithm, and b) Key. Sometime, multiple keys can also be used for encryption. A number of cryptographic algorithms are available in market such as DES, AES, TDES and RSA. The strength of these encryption algorithms depends upon their key strength. Strong encryption algorithms and optimized key management techniques always help in achieving confidentiality, authentication and integrity of data and reduce the overheads of the system.

Cryptography is basically divided into two categories [2]; a) Symmetric Cryptography, and b) Asymmetric Cryptography. In symmetric cryptography the key used to encrypt the message is the same as the key decrypting the message whereas in asymmetric

Cryptography different key is used for encryption and decryption.

The work presented in this paper aims at the following aspects.

• Develop a new hybrid technique for improving the security using encryption and decryption algorithms.
• Compare the various techniques at hand with the proposed system.

• Build a system that delivers optimal performance both in terms of speed and accuracy.

Overview Of Work

Particle The core of Hill-cipher [3] is matrix manipulations. It is a multi-letter cipher, developed by the mathematician Lester Hill in 1929. For encryption, algorithm takes m successive plaintext letters and instead of that substitutes m cipher letters. In Hill cipher each character is assigned a numerical value like:

a = 0,
b = 1,
…..
…..
z = 25.

The substitution of cipher text letters in place of plaintext leads to m linear equations. For m = 3, the system can be described as follows:

$$C_1 = (K_{11}P_1 + K_{12}P_2 + K_{13}P_3) \text{ MOD } 26$$
$$C_1 = (K_{21}P_1 + K_{22}P_2 + K_{23}P_3) \text{ MOD } 26$$
$$C_1 = (K_{31}P_1 + K_{32}P_2 + K_{33}P_3) \text{ MOD } 26$$

This can be expressed in terms of column vectors and matrices:

$$C = KP$$

Where C and P are column vectors of length 3, representing the plaintext and the cipher text and K is a 3*3 matrix, which is the encryption key. All operations are performed mod 26 here. Decryption requires the inverse of matrix K. The inverse K^{-1} of a matrix K is defined by the equation. K K^{-1}= I where I is the Identity matrix.

K^{-1} is applied to the cipher text, and then the plain text is recovered. In general terms we can write as follows:

For encryption: $C = E_k(P) = Kp$

For decryption: $P = D_k(C) = K^{-1} C = K^{-1}Kp = P$

2 PROPOSED WORK

As we have seen in Hill cipher decryption, it requires the inverse of a matrix. So while one problem arises that is: Inverse of the matrix doesn't always exist. Then if the matrix is not invertible then encrypted text cannot be decrypted.

In order to overcome this problem author suggests the 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.

The Modification in Hill cipher algorithm generates the different key matrix for each block encryption instead of keeping the key matrix constant. It increases the secrecy of data and algorithm also checks the matrix used for encrypting the plaintext whether that is invertible or not. If the encryption matrix is not invertible, the algorithm modifies the matrix such a way that it's inverse exist. The new matrix obtained after modification of key matrix is called known as Encryption matrix. In order to generate different key matrix each time the encryption algorithm randomly generates the seed number and from this key matrix is generated [6][7].

Key matrix,

$$K = \begin{bmatrix} K_{11} & K_{12} K_{13} \\ K_{21} & K_{22} K_{23} \\ K_{31} & K_{32} K_{33} \end{bmatrix}$$

Where,

$$K_{11} = \text{seed number}$$

$$K_{12} = (\text{seed number} * m) \bmod n$$

$$K_{11} = (12K * m) \bmod n$$

$$K_{11} = (13K * m) \bmod n$$

Where m is successive numbers of plaintext letters taken at a time for encryption and 'n' is length of the lookup table or we can set this 'n' value as per requirement. Then with the help of key matrix encryption matrix 'E' is generated. For self repetitive matrix, matrix should be square and it should be non-singular.

2.1 Generation of a self repetitive Matrix for an 'n'

If the matrix is of dimension greater than and with mod index greater than 91, the methods of brute force are not performed. It takes very long time and 'n' value may be in the range of millions and 'n' is the value where the matrix becomes an identity matrix. If the computations will be matrixes or more a normal Pentium 4 machine takes more processing time.

Hence, it would be comfortable to know the value of and then generate a random matrix. 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, ... and Now taking the LCM of these values gives 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 'E'.
4. Generate $c = E^{-1}AE$
5. The 'n' value of 'C' is also 'n'

2.2 Mathematical proof generation of a *self repetitive matrix* for an 'n'

$$(E^{-1}\, AK)n = (E^{-1})\, n * (A)\, n * (E)\, n$$

AN = I as calculated before as it is a diagonal matrix and 'n' is the LCM of all elements

$$(E^{-1}\, E) * (E^{-1} * E) \ldots\ldots n \text{ times} = I$$

2.3 Cipher text Development

First take plaintext and represent this in the form of a matrix, given by

B = input ('Enter the block of string')

$$P = [pij\,], I = 1 \text{ to } n, j = 1 \text{ to } n. \text{ (Public key)}$$

Let us choose a secret key matrix K,

$$K = [kij\,], I = 1 \text{ to } n, j = 1 \text{ to } n,$$

and

$$E = [\,eij], i = 1 \text{ to } n, j = 1 \text{ to } n,$$

DOI: 10.18535/ijecs/v5i10.07

Obtained by key matrix an increments in diagonals element in K

Here, we assume that the determinant of E is not equal to zero and it is an odd number. In view of this fact the modular arithmetic inverse of E can be obtained by using the relation

$$(EE-1)MOD97 = I$$

On assuming that e_{ij} the elements of the matrix E are odd numbers lying in [1-97], we get the decryption key matrix E-1 in the form

$$E^{-1} = Inv\,[E],$$

Where e_{ij} and d_{ij} are governed by the relation

$$(eij \times dij)\,mod\,97 = 1$$

Here, it is to be noted that d_{ij} also turn out to be odd numbers in [1-97]. The basic equations governing the encryption and the decryption are given by

$$P = (pij)$$

$$E = [eij \times pij]\,mod\,97, i = 1\ to\ n, j = 1\ to\ n,$$

$$C = E * B$$

and

$$C = [cij] = [dij \times cij]\,mod\,97, i = 1\ to\ n, j = 1\ to\ n$$

$$P = (E^{-1}C)\,mod\,97.$$

The corresponding algorithms for the encryption and the decryption are as follows.

2.4 Algorithm for Encryption

1. $Read\ B, P, E, K, n, r$

2. $For\ k = 1\ to\ r\ do$

{

3. $P = pij$

4. $For\ i = 1\ to\ n\ do$

5. $E = eij$

6. $For\ j = 1\ to\ n\ do$

{

7. $E = (pij \times eij)\,mod97$

} }

8. $C = [E * B]$

}

9. $C = [cij]$

10. $Write\ (C)$

2.5 Algorithm for Decryption

1. $Read\ C, E, K, n, r$

2. $E^{-1} = Inv\,(E)$

3. $For\ k = 1\ to\ r\ do$

{

4. $C = [cij]$

5. $B = E^{-1}C)\ mod\ 97$

}

6. $Write\ (B)$

2.6. Flowchart for Encryption & Decryption

Figure 1 shows the flow chart for the algorithm of encryption and decryption using modified Hill – Cipher method.

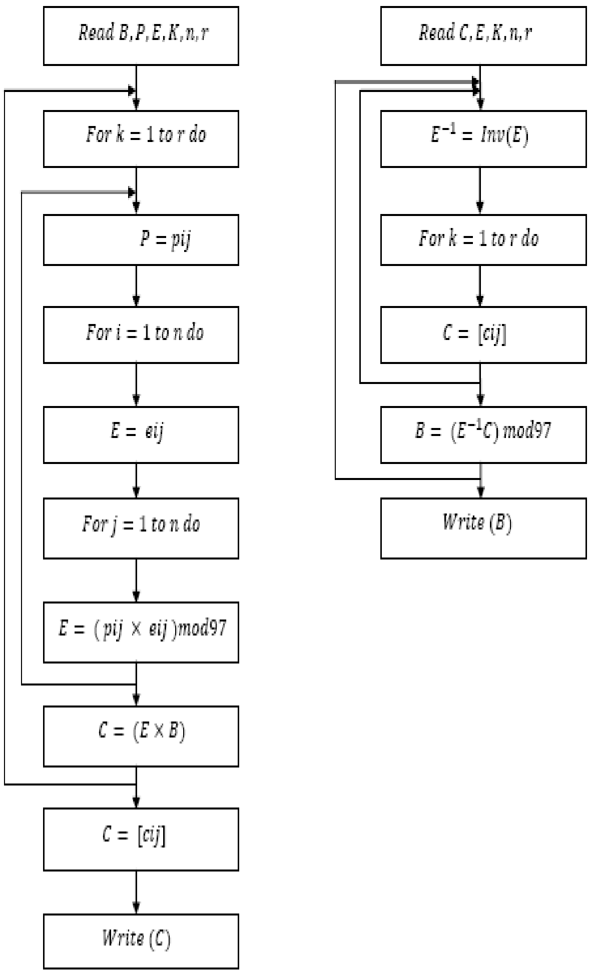

Figure 1: Flow chart of Encryption and Decryption Algorithm

3 IMPLEMENTATION & RESULTS

Performance measurement criteria are time taken by the algorithms to perform the encryption and decryption of the

DOI: 10.18535/ijecs/v5i10.07

input text file that is encryption computation time and decryption computation time.

3.1 Encryption Computation Time

The encryption computation time is the time which is taken by the algorithms to produce the cipher text from the plain text. The encryption time can be used to calculate the encryption throughput of the algorithms.

Table 1 below shows Encryption Execution Time for Different File Sizes. For the file of 10Kb in size the encryption execution time for original Hill cipher, Modified Hill cipher and proposed algorithm are 13, 8 and 5 msec respectively and for file size of 100 kb the encryption execution time are 86, 62 and 48 msec respectively. It is shown that proposed algorithm consumes less time for all types of file sizes.

Table 1: Encryption Execution Time for Different File Sizes

Input File	**Original Hill Cipher**	**Modified Hill Cipher**	**Proposed Algorithm**
File Size (Kb)	Encryption Execution time (msec)	Encryption Execution time (msec)	Encryption Execution time (msec)
10	13	8	5
20	17	13	11
30	25	19	15
40	21	22	17
Total Size 100 Kb	**86 msec**	**82 msec**	**48 sec**

3.2 Decryption Computation Time

The decryption computation time is the time taken by the algorithms to produce the plain text from the cipher text. The decryption time can be used to calculate the decryption throughput of the algorithms.

Table 5.2 below shows Decryption Execution Time for Different File Sizes. For file size of 100Kb the decryption execution time are 109, 91 and 74 msec respectively. It is shown that proposed algorithm consumes less time for all types of file sizes.

Table 5.2: Decryption Execution Time for Different File Sizes

Input File	**Original Hill Cipher**	**Modified Hill Cipher**	**Proposed Algorithm**
File Size (Kb)	Decryption Execution time (msec)	Decryption Execution time (msec)	Decryption Execution time(msec)
10	17	15	9
20	22	18	16
30	31	25	21
40	39	33	28
Total Size 100 Kb	**109 msec**	**91 msec**	**74 sec**

3.3 Results

From the simulation result, it shows that when the cipher text is decrypted with the help of public or private keys we get the same plaintext. It can be observed that proposed approach provides less commutation time for all types of file sizes when compared to other algorithms. The proposed algorithm is optimized compared to other algorithms in terms of hacking and processing time. So the accuracy and secrecy of proposed algorithm is better than other existing algorithms.

4. CONCLUSION

Cryptography provides solution for data integrity, authentication and non-reproduction. The Hill cipher technique using a novel method of self-repetitive matrix and it has been successfully implemented. From the experimental results it has been shown that the modified Hill Cipher is easy to implement and difficult to crack. This technique becomes more secure by using modular arithmetic. The block size which is specified as 64 bit is expandable as per requirement, thus gives flexibility in message string length. It generates key of 56 bits which is enhance the security aspect of this algorithm and make them more secure than other encryption algorithms. Due to the following facts it has been concluded that it takes very less time for execution as compare to other Hill Cipher algorithm.

Using the Hill Cipher, performance will be appropriate in much kind of applications where it is suitable. The proposed algorithm has been compared with other algorithms and found that throughput of proposed algorithm is greater than other encryption algorithms. Future work will be carried out to decrease the complexity of the proposed algorithm.

REFERENCES

[1] W. Stallings; “Cryptography and Network Security” 2nd Edition, Prentice Hall,1999

[2] Bruce Schneir: Applied Cryptography, 2nd edition, John Wiley & Sons, 1996

[3] M. Alqdah and L.Y. Hui, “Simple Encryption and Decryption Algorithm” International Journal of Computer Science and Security, Vol. 1, pp. 14-17, 2008.

[4] A. Kakkar, P.K. Bansal, “Reliable Encryption Algorithm used for Communication”, M. E. Thesis, Thapar University, 2004.

[5] D. R. Stinson, “Cryptography Theory and Practice”, 3rd edition Chapman Hall, Vol. 1, pp. 13-37, 2006.

[6] Lester. S. Hill, “Cryptography in an algebraic alphabet”, Amer. Math, Vol. 1, pp. 306-312, 1936.

[7] B. Acharya, S. K. Panigrahy, S. K. Patra and G. Panda, "Image Encryption Using Advanced Hill Cipher Algorithm", International Journal of Recent Trends in Engineering, Vol. 1, No. 1, pp. 663-667, 2009.

[8] D. Anand, V. Khemchandani and R. K. Sharma, "Identity Based Cryptography Techniques and Applications", International Conference on Computational Intelligence and Communication Networks, Vol. 1, pp. 343-348, 2013.

[9] S. F. Mare, M. Vladutiu and L. Prodan, "Secret data communication system using Steganography, AES and RSA", International Symposium for Design and Technology in Electronic Packaging, Vol. 2, pp. 339-344, 2011.

[10] S. Khar, N. Bhargawa, R. Shukla and M. Shukla, "Implementation and Enhanced Modified Hill Cipher by P-box and M-box technique", International Journal of Information Technology and Knowledge Management , Vol. 5, No.1, pp. 53-58, 2012.

[11] Pavan. N, Nagarjun G. A, Nihaar N, G. S Gaonkar and P. Sharma, "Image Steganography Based On Hill Cipher with Key Hiding Technique", IOSR Journal of Computer Engineering, Vol. 11, pp. 47-50, 2013.

[12] K. R. Saraf, V. P. Jagtap, A. K. Mishra, "Text and Image Encryption Decryption Using Advanced Encryption Standard", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS) Volume 3, Issue 3, May – June 2014

[13] S. Desai, C. A. Mudholkar, R. Khade, P. Chilwant, "Image Encryption and Decryption using Blowfish Algorithm", International Journal of Electrical and Electronics Engineers IJEEE, Volume 07, Issue 01, June 2015.