International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 03 | Mar 2022 www.irjet.net p-ISSN: 2395-0072

ELECTRONIC MAIL SECURITY USING ASYMMETRIC CRYPTOGRAPHIC ALGORITHM: A NOVEL APPROACH

Akkimsetti Purna Veera Manindra

Dept. of Computer Science Engineering, School of SOC SASTRA University, Thanjavur, Tamil Nadu, INDIA

Abstract - *Electronic mail contains confidential information such as transaction details, fund transfer information, personal details, security numbers, etc. There is a high chance that the data can be well exposed during transmission over the internet, so there exists always the possibility of theft or manipulating the data. In this paper, A new mathematical algorithm of public-key encryption which encrypts the data using mathematical concepts like matrix properties is developed and analyzed to ensure secure electronic mail communication. This algorithm enciphers the email data into a matrix and retrieves the original data by decrypting the encrypted data. The performance of algorithms and data ciphering is also evaluated*.

***Key Words*: - Electronic mail, enciphering, deciphering, Matrix, key generation**

1. INTRODUCTION

In the modern-day world, E-Mail has become the most common way of communication between the user and clients over public channels. With an exponential increase in emails, there is a high risk of getting into unintended recipients. Cryptography can be defined as a methodology that ciphers the data, using different mathematical algorithms that make the data indecipherable unless deciphered by an intended user. By default, emails are not end-to-end encrypted [9].

There are majorly two types of ciphering algorithms, symmetric and asymmetric ciphering. In symmetric ciphering keys for both encryption and decryption processes are the same whereas asymmetric ciphering of different keys are used i.e., the public key and private key where both are related mathematically [7]. Asymmetric encryption is also known as public-key encryption. Block cipher and Stream cipher are most popular in data encryption affiliated to Symmetric key cipher. Block ciphers were first developed by Claude Shannon in 1949 [17], improving security using replacements and permutations using 64 bits. It works on transposition approaches like columnar transposition technique, rail-fence cipher. Electronic Code Book (ECB), Cipher Block Chaining (CBC) are the algorithm modes used by block cipher [18]. Gilbert Sandford Vernam 1917, developed an additive polyalphabetic stream cipher in which key is combined character by character with the text to produce the ciphertext [19]. It toils on substitutions approaches like Polygram substitution cipher and Caesar cipher. Cipher Feedback and Output Feedback are some algorithms used in stream ciphers. Twofish algorithm uses symmetric encryption, considered as one of the fastest ciphering algorithms, requires 256-bit keys for encryption used for both Software and Hardware environments.

Diffie-Hellman key exchange was one of the popular key exchange protocols is a method of exchanging the public keys over a public server as it is a non-authenticated key agreement protocol ,used to serve forward secrecy[8], Elliptical curve cryptograph (ECC) is a key based technique for ciphering the data often discussed in the context of RSA Algorithm developed in 1977 [12], Rivest – Shamir-Adleman (RSA) one way enciphering of things like electronic-mail, data, software using factorization, El Gamal, DSA developed by Ron Rivest and Adi Shamir et al [13] are some encryption algorithms. Apart from these another popular technique is hashing developed by Ron Rivest in 1989. A unique way of ciphering which is different from encryption and decryption. Hashing is an irreversible process (the original data cannot be retrieved back), uses a hash algorithm to store the data with a unique hash value .it plays an important role in Banking applications like password management and other block chain applications [20].

A SEGPX distributed search engine was developed by Lu and Geva [18] for secure email communication. It uses X.50 public key attribute frameworks which utilizes email channels for data transmission. Jain, Gosavi [22] encrypted using variable key and compress the data using code book, compressed messages must be encrypted to maintain secure email. With compression of messages the overload of traffic over the channel will get reduced. A proxy based secure electronic mail system using IBE Cryptography was developed by Chen and Ma [11]. In Linux platform for secure email communication Madi [20] proposed a simple implementation that uses local based email transfer agent. To improve security in email infrastructure Choukse and others explain weakness in immanent methodologies and highlighted good and weak practices in email design. A point-to-point email ciphering protocol and an identity based on one-way key agreement was introduced by Yeh [26]. For secure email communication Ojamaa and Lind [21] developed a solution based on public key ciphering and OpenPGP to the users. Hill cipher developed a substitution cipher in 1929, in which each element in the text is substituted with a number in Z_{26} [14]. Leon Battista Alberti ,1947 developed a polyalphabetic cipher for

encryption in which text is segregated and then substituted with suitable keys. He also presented Cipher disk for encryption. Vigenère cipher also uses the simple form of polyalphabetic substitution, encryption is done using Vigenère table in which letters written out 26 times in divergent rows.

Public key cryptography was developed by Martin Hellman, Ralph Merkle, and Whitfield Diffie at Stanford University in 1976 [16]. In Public key encryption, public key is public to everyone and private key belongs to the recipient who create these keys. Initially recipient generates the public and private key pair in which public key public to everyone and the sender use the public key for encryption and send the data over the server, on the other end receiver uses the private key for decipher the data. Authentication helps in securing the keys from hackers.

End-to end encryption uses public key cryptography as data is enciphered and deciphered only at the end points. when the sent an email with an it enciphered at the source and again it deciphered at its end point, and it is unreadable in during transmission in public servers [10]. The prime solutions are available now that provides end to end encryption for electronic mail are Secure Internet Mail Extension (SIME), pretty good privacy (PGP), are using public key encryption [11].

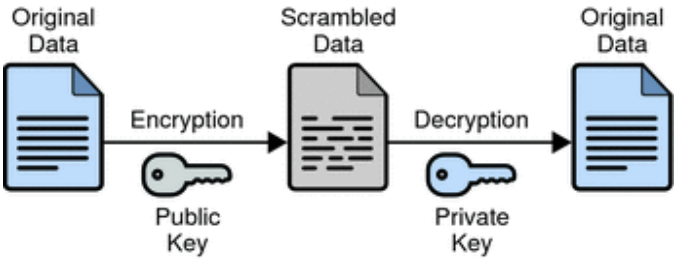


Fig 1. working of public key encryption and decryption

In this paper a new key generation algorithm in public key encryption is developed and analyzed to encrypt and decrypt the data.

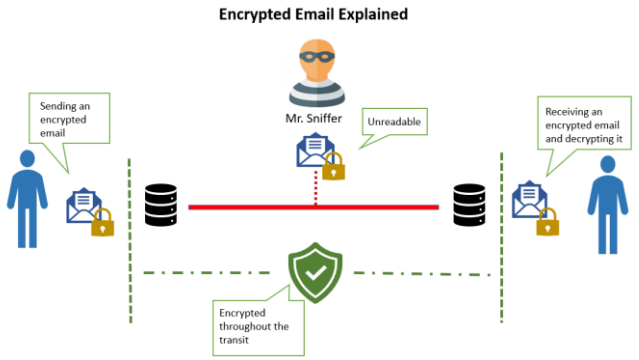


Fig 2. E-mail transmission through public server

2. EXIGENCY OF EMAIL CIPHERING:

Since our sensitive information is being communicated through email, it's a cardinal thing to secure the data. In recent years, security has become the main concern for everyone, there are so many ways for a third-party recipient for bypassing the data. Attacks like phishing, Man-in-the-middle are the most common email hacking techniques and other attacks like Spyware, Spam, Adware, are some E-mail attacks [14], and also E-mails exposed to both passive and active threats. passive threats like traffic analysis, snooping and exposing the data while the active threats include mitigating the data and Denial of service (DoS & DDoS) etc. So, the data should be protected from being exposed to the attackers So, clearly there is a need of ciphering the data from its original form. Though there so many advanced algorithms exist today for encryption, but for any algorithm we have input the key in order to cipher the data. But in case of emails declaring keys every time is not possible.in this algorithm with the key generation algorithm random keys will generate automatically.so the emails get enciphered and deciphered automatically without user interference.

3. ALGORITHM

Mathematics plays important role in Cryptography. We can say Cryptography is the science of using mathematics to camouflage the data behind cyphertext. It entails stowing secret information with a pair of public key and private key that recipient must have, so has to access the original message.[4]

Using matrices for cyphering is a distinctive way in the field of cryptography [6]. This algorithm is related to mathematics in which matrix concepts are used. every element in the electronic mail is ciphered into numerical value based on the (American standard code for information interchange) ASCII values.

The numerical values will be stowed in the N*N message matrix where the order of N depends on the size of the electronic mail.

Order of the matrix=√Size of the data + 1

The Matrix should be a square matrix hence the remaining elements in the message matrix is filled with **ASCII value 32 i.e., Blank Space.** This process of adding blank space is called padding.

3.1 Key Generation Algorithm

$$[K]^{-1} = \text{Adjoint}(K) * 1/|K|$$

[K]=Matrix of order N

|K|=determinant of matrix

Adjoint(K)=Adjoint matrix of K

This equivalent to,

|K|=[K]*Adjoint(K)

We will obtain the private key using the determinant value that is K.

From the properties of matrix, **the determinant of a lower triangular matrix (or an upper triangular matrix) is the product of the diagonal entries.** [6] fact7

Structure of private Key Matrix: -

X	**0**	**0**	**0**	**0**
D	**1**	**0**	**0**	**0**
D	**D**	**1**	**0**	**0**
D	**D**	**D**	**1**	**0**
D	**D**	**D**	**D**	**1**

Here X represent Determinant value which is the public key

D is Don't care (Any random number)

We can arrange the diagonal elements differently in such a way that the sum of all elements in the diagonal should equal to the determinant value.

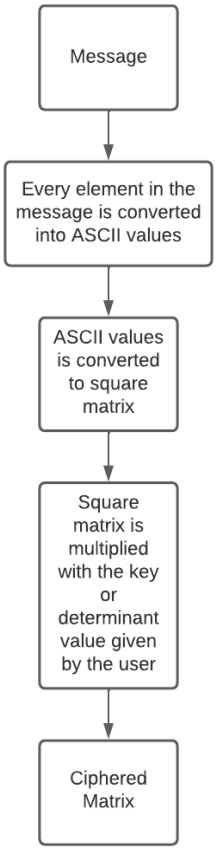
This algorithm produces a Matrix whose determinant value is equal to public key. And this secret key is used for deciphering by the recipient

3.2Enciphering Algorithm

[Ciphered Matrix] = [Message Matrix] *|K|

In this Enciphering Algorithm we multiply the Message ASCII matrix with the determinant value that is public key. The obtained matrix is the encrypted matrix from the email data. Public key is multiplied with identity matrix which is used a key for encryption

Flow Diagram for Enciphering

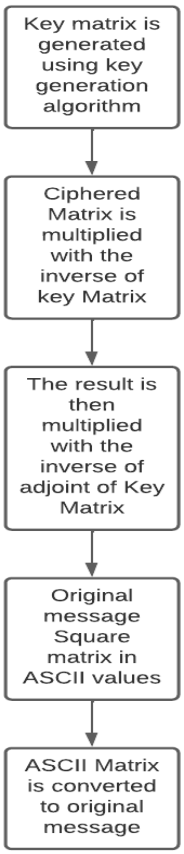


3.3 Deciphering Algorithm: -

[Message Matrix] = [ciphered Matrix] *[Key Matrix] $^{-1}$ * [Adjoint (Key Matrix)] $^{-1}$

In deciphering Algorithm to obtain the message matrix from its enciphered matrix we should multiply the ciphered matrix with the inverse of key matrix and with the inverse of adjoint key matrix in which the key matrix is obtained by the key generation algorithm with the help of the determinant value. We used different key value to enciphering and deciphering So this is public key encryption.

Flow Diagram for Deciphering



Precedent:

Let us deliberate an example with enciphering and deciphering a message "**E-Mail", the** size of this message is 6 so the minimum square matrix required is 3*3 the message is converted to ASCII matrix as follows and the remaining 3 blocks in matrix is filled with ASCII value 32 that is Blank Space.

$$\text{ASCII Matrix} = \begin{bmatrix} 69 & 45 & 77 \\ 97 & 105 & 108 \\ 32 & 32 & 32 \end{bmatrix}$$

Let us consider the pubic key value also determinant value as 23

Ciphered matrix= [ASCII matrix] *(key)

$$\text{Ciphered Matrix} = \begin{bmatrix} 69 & 45 & 77 \\ 97 & 105 & 108 \\ 32 & 32 & 32 \end{bmatrix} * \begin{bmatrix} 23 & 0 & 0 \\ 0 & 23 & 0 \\ 0 & 0 & 23 \end{bmatrix}$$

$$\text{Ciphered Matrix} = \begin{bmatrix} 1587 & 1035 & 1771 \\ 2231 & 2415 & 2485 \\ 736 & 736 & 736 \end{bmatrix}$$

Obtaining key matrix from its determinant value using key generation Algorithm

$$\text{Key matrix} = \begin{bmatrix} 23 & 0 & 0 \\ 0.2382 & 1 & 0 \\ 0.4605 & 0.6081 & 1 \end{bmatrix}$$

$$(\text{Key Matrix})^{-1} = \begin{bmatrix} 0.04347826 & 0 & 0 \\ -0.02950619 & 1 & 0 \\ 0.00185449 & -0.34043 & 1 \end{bmatrix}$$

$$(\text{Adjoint (Key Matrix)})^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0.00750814 & 0.04347826 & 0 \\ 0.04018938 & 0.01216355 & 0.04347826 \end{bmatrix}$$

Deciphered Matrix= [ciphered Matrix] * $(\text{Key Matrix})^{-1}$ * $(\text{Adjoint (Key Matrix)})^{-1}$

$$\begin{bmatrix} 1587 & 1035 & 1771 \\ 2231 & 2415 & 2485 \\ 736 & 736 & 736 \end{bmatrix} * \begin{bmatrix} 0.04347826 & 0 & 0 \\ -0.02950619 & 1 & 0 \\ 0.00185449 & -0.34043 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 \\ 0.00750814 & 0.04347826 & 0 \\ 0.04018938 & 0.01216355 & 0.04347826 \end{bmatrix}$$

$$\text{Deciphered Matrix} = \begin{bmatrix} 69 & 45 & 77 \\ 97 & 105 & 108 \\ 32 & 32 & 32 \end{bmatrix}$$

Message=" E-Mail"

This deciphered matrix is converted to message using ASCII chart and the original message is obtained

3.4 Algorithm Implementation

```
1. x=input ("input E-mail message:")
2. Key=int (input ("input Public Key:"))
3. final= []
4. ascii= []
5. messageKey= []
6. for character in x:
7. ascii. append(ord(character))
8. size=math. floor (math. sqrt(len(ascii))) +1
9. messageKey = [element * Key for element in ascii]
```

```
n, m = size, size
k = 0
msgMatrix = []
while n*m! = len(ascii):
ascii. append (32)
for idx in range (0, n):
sub = []
for jdx in range (0, m):
sub. append(ascii[k])
k += 1
msgMatrix.append(sub)
while n*m! = len(messageKey):
messageKey.append(32*Key)
n, m = size, size
l=0
keyMsg= []
for idx1 in range (0, n):
sub1 = []
for jdx1 in range (0, m):
sub1.append(messageKey[l])
l += 1
keyMsg.append(sub1)
print ("Enciphered Matrix")
print(keyMsg)
```

Code 1. Message Encryption Implementation

```
1. random_matrix = [[random. random () for e in range(len(keyMsg))] for e in range(len(keyMsg))]
2. convert random matrix into lower triangular matrix
3. Assign all diagonal elements in random matrix to 1
4. random_matrix [0][0] =Key
5. det = np.linalg.det(random_matrix)
6. round(det)
7. adj_random= []
8. inverse_random_Matrix= []
9. inverse_adj_random= []
10. adj_random=adjoint_matrix(random_matrix)
11. inverse_random_Matrix=np.linalg.inv(random_matrix)
12. inverse_adj_random=np.linalg.inv(adj_random)
13. Product_rand_adjrand=[]
14. Product_rand_adjrand=np.dot(inverse_random_Matrix,inverse_adj_random)
15. decoded_matrix= []
16. decoded_matrix=np.dot (keyMsg, Product_rand_adjrand)
17. dec_list=[]
18. dec_list=decoded_matrix. flatten ()
19. dec_list = [int(round(x)) for x in dec_list]
20. print(dec_list)
21. decrepted_Message = "". join([chr(value) for value in dec_list])
22. final. append(decrepted_Message)
23. print ("deciphered message:")
24. print ('. join(final))
```

Code 2. Message Decryption Implementation

The above code is developed using Visual studio code, python version 3.9.7. Following is the output of the above implemented code

```
PROBLEMS 27  OUTPUT  DEBUG CONSOLE  TERMINAL

(BM_1301) akkimsettipurnaveeramanindra@Akkimsettis
-MacBook-Air C++ codes % python -u "/Users/akkimse
ttipurnaveeramanindra/Documents/C++ codes/final_pa
per.py"
input E-mail message :E-mail Ciphering

input Public Key :5678


size of the matrix : 5

Message Matrix

[[69, 45, 109, 97, 105], [108, 32, 67, 105, 112],
[104, 101, 114, 105, 110], [103, 32, 32, 32, 32],
[32, 32, 32, 32, 32]]


Enciphered Matrix

[[391782, 255510, 618902, 550766, 596190], [613224
, 181696, 380426, 596190, 635936], [590512, 573478
, 647292, 596190, 624580], [584834, 181696, 181696
, 181696, 181696], [181696, 181696, 181696, 181696
, 181696]]


*******Deciphering ******

decipheed Matrix

[[ 69.  45. 109.  97. 105.]
 [108.  32.  67. 105. 112.]
 [104. 101. 114. 105. 110.]
 [103.  32.  32.  32.  32.]
 [ 32.  32.  32.  32.  32.]]

[69, 45, 109, 97, 105, 108, 32, 67, 105, 112, 104,
 101, 114, 105, 110, 103, 32, 32, 32, 32, 32, 32,
32, 32, 32]

decihered message :

E-mail Ciphering

Time:  5.119090916999999
```

Size of the Message	Key	Execution time
5	1234	7.271491708
29	98754	10.50398475
64	123456789098765432	9.574130458
131	78	10.517340625000001
395	3456	7.507566458

Table1. Execution time taken for above code with different inputs

Results:

With the help of this algorithm, we enciphered and deciphered the electronic mail data. Initially, we enciphered the message into ASCII matrix with the help of ASCII table whose order depends on the size of the data given as input and to encrypt the message, the ASCII Matrix is multiplied by a public key generated by the key generation algorithm. for obtaining the original data from enciphered matrix we used the deciphering algorithm and key generation algorithm which is the only way to decipher the data from its enciphered one.as the key generation is random, and also it uses different algorithms for encrypting and decrypting it is impossible to predict by any 3rd person. In email we integrate the encryption algorithm in the backend of the client as the public key from the server will be available publicly to everyone and for decryption integration is done at the server side which uses the private key of the server to retrieve the data.

Discussions:

The electronic mail ciphering should be done instinctively without user and recipient intervene and also mails should get ciphered swiftly to shun time delay. Now a days no one wants to send uncyphered electronic email due to variety of threats. As there are so many state-of-the-art algorithms these days but many of these are complex and time taking process as the ciphered text is larger than the original message. but in this algorithm the size of the ciphered matrix depends on the given input so this is fast enough when compared to other algorithms. Maximum size of electronic mail is 25 MB so it can be handled easily by modern devices and we are using ASCII values as they are valid for all characters and we are using key generation algorithm for enciphering and deciphering we get a random key every time also the size of the key is differing from data to data. As there are 2 different keys as public and private key for enciphering and deciphering produced by itself provides extra salvation.

Conclusion:

In this paper a modified version of public key encryption has been developed and analyzed for enciphering and deciphering the data. A unique technique /algorithm to encipher and decipher the material. Since there is a possibility of electronic mail hacking techniques like keylogging and phishing this algorithm Avoid those things with key generation algorithm and as it included of matrix inversion and multiplication concepts and other mathematics applications, hacking the data is very hard.in Cryptographic process among all the methods using matrices is the strongest method because it uses mathematical concepts. An additional layer of security will be added to the data by encrypting and decrypting it.

References:

1- “Email Encryption using RC4 algorithm” in proceeding of Meltem Kurth Pehlivanoglu and Nevcihan Duru in November 2015 International Journal of Computer Applications

2- “Public Key Cryptography” by Dwi Liestyowati in proceedings of Journal of physics 2019

3- “The Role of Mathematics in Emergence of Cryptography” by Savkirat Kaur in proceedings of International Journal of Advanced Research in Science and Engineering

4- image 1 source from: https://privacycanada.net/mathematics/

5.image 2 source from - https://docs.oracle.com/cd/E19656-01/821-1507/aakfv/index.html

6- “Symmetric and Asymmetric Encryption” by Gustavus j. Simmon proceeding of Priceton university

7-” Secure Communication over insecure Channels” by Ralph C and Merkle in April 1978 proceeding of communication of the ACM

8- "Enable hosted S/MIME for enhanced message security". GSuite Admin Help. Google. Retrieved 2020-06-15.

9- "End-to-end encryption". How To Geek. Retrieved 9 April 2015.

10- “Network Security Applications and Standards” by W. Stallings in 2000, Prentice Hall.

11- “Public key Cryptography” by S L Garfinkel Oreilly and Associates in proceeding of Institute of Electrical and Electronics Engineers (IEEE) in 1996

12- “Hill Cipher and Vigenère Cipher” by Munir, Rinaldi, Dikat in 2006 IF5054 Cryptography

13-” classification of Email Attacks” Link- https://www.geeksforgeeks.org/types-of-email-attacks/

14- "Communication Theory of Secrecy Systems" by Shannon, Claude, 1949 in Bell System Technical Journal

15- “Securing Record Communication” by Klein, Melville in October 2012 retrieved in April 2012.

16-Ojamaa, A. and Lind, U. R. 2013. Securing Customer Email Communication in E-Commerce. Sixth International Conference on Developments in E-Systems Engineering.

17- Email Security using Encryption and Compression. Computational Intelligence for Automation International Conference & Modelling Control by Yogendra Kumar Jain Pramod B. Gosavi in proceedings of IEEE 2008.

18-Secure email-based peer to peer information retrieval, (ICC) International Conference on Cyberworlds by Lu, C. and Geva, S in 2005.

19-A Secure Email Encryption Proxy Based on Identity-based Cryptography by T Chen and Shilong Ma in 2008. And conference on Information Technology and Multimedia

20-Implementation of Secure Email Server in Cloud Environment. Conference on Computer and Communication Engineering (ICCCE) by Madi, NKM., Solmaz,S., Farzaneh,M., and Azizol, A in 2012.