A Secured Message Transaction Approach by Dynamic Hill Cipher Generation and Digest Concatenation

Md. Saiful Islam Chowdhury, Shoyeb Al Mamun Shohag, Md. Hasan Sahid
MS in Telecommunication, Dept. of Computer Science & Engineering
University of Information Technology & Sciences (UITS)
Baridhara View, GA 37/1, Dhaka-1212, Bangladesh

ABSTRACT

An electronic transaction system is a set of participants and their interactions towards an efficient and secured exchange of message between the participants. A secured electronic message transaction system has been designed, developed and implemented where Hill Cipher Cryptosystem is used as the encryption-decryption process with dynamic keys. For this process, any transacted message is to be inputted into the proposed system, and then a matrix of the message length is calculated with its inverse matrix. The calculated inverse matrix is sent to the receiver where he/she used it as the key for the cryptosystem. The key matrix and the Hill Cipher technique have been used to generate the cipher text. The message digest algorithm MD5 operations have been used on the message to generate the digest of the message. The generated digest is concatenated with the encrypted message and is sent to the destination. In the receiver end, the intended receiver first computes the digest of the message by performing the message digest algorithm MD5 and compares it with the received digest that establishes the data integration and non-repudiation of the proposed system. The receiver then decrypts the received encrypted message using the Hill Cipher technique with the received inverse matrix and retrieve the message properly which establishes the message confidentiality and authentication. The proposed system has been implemented using the Java programming and analyzed for several applications. Finally, the fundamental security services have been measured and analyzed. This may be applicable for cryptographic applications.

General Terms

Network Security

Keywords

Cryptography, MD5, Digital signature, Hill cipher technique, Plain texts, Cipher text, RSA, FDH, DSA, MD4, ASCII, MAC, IP, HASH, ID.

1. INTRODUCTION

Cryptography is a mechanism that is used to design and implement secure electronic message transaction system, where plain texts are transmit after encryption with the encryption algorithm. The encryption algorithm forms with secret key value and an algorithm, is used to produce the cipher text that is to be sent to the destination. The receiving end, to produce a plain text from the received cipher text, decryption algorithm and decryption key is used. Symmetric key consist same values in both sender and receiver side that is used for encryption of plain text and for decryption of cipher text. Asymmetric key consist different values in both sender and receiver side. Different symmetric-key cryptographic techniques and message authentication processes are employed to design a secured message transaction system. With this paper, Hill Cipher generation technique with dynamic key generations are considered for designing the cryptographic system and a simple but strong secured message authentication for implementing the digit concatenation. Secured electronic message transaction system requires the key generation, key distribution and management, and the security services established by the session-keys. Electronic message transaction system is used to transform information over the internet between communication parties as an electronic form. For secure transaction over the internet, secret key crypto system is used to build up the basic building blocks [1][9].

The next part of this paper is structured as follows. A background study has been point out in section 2. In section 3, Present conventional message transaction approaches have been presented. Section 4 and 5 holds the methodology and implementation of our proposed secure message transaction system. Security analysis of our implemented system has been done in section 6. Conclusions and future works have been presented in section 8 and 9.

2. BACKGROUND STUDY

In cryptography, digital signature scheme is used to identify the security properties of a handwritten signature on paper. To build up a digital signature at least three algorithms need to have those key generation algorithm, a signature algorithm and a verification algorithm. Key generation algorithm selects a private key uniform at random, from a set of possible private keys and the algorithm outputs the private key and a corresponding public key. A signature algorithm which given a message, a private key and produces a signature. A verification algorithm which given a message, public key and a signature, either accepts or rejects. It gives two benefits, those are Authentication, which is used to identify the message originator and Integrity that gives the opportunity to determine if any alteration happens during the transmission of messages between communication parties. There are many kind of digital signature algorithm and they are applicable with variety of applications. Few of those are RSA (Rivest-Samir-Adleman algorithm), FDH (Full Domain Hash), DSA (Digital signature algorithm), SHA (Secure hash algorithm), Aggregate Signature, and Schnorr Signature [8].

MD5 (Message-Digest algorithm 5) is a cryptography hash function and works with 128-bit hash value. Ron Rivest, who has been designed this algorithm in 1991 to replace an earlier hash function, MD4 (Message-digest protocol 4). It is a widely used algorithm in cryptography and often used to check the integrity of a file. An MD5 hash is typically expressed as a 32 digit hexa decimal number. MD5 processes a variable-length message into a fixed-length output of 128 bits. The input message is broken up into chunks of 512-bit blocks (sixteen 32-bit little endian integers); the message is padded so that its length is divisible by 512. The padding works as follows: first a single bit, 1, is appended to the end of the message. This is followed by as many zeros as are required to bring the length of the message up to 64 bits less than a multiple of 512. The remaining bits are filled up with a 64-bit integer representing the length of the original message, in bits. The main MD5 algorithm operates on a 128-bit state, divided into four 32-bit words, denoted A, B, C and D. These are initialized to certain fixed constants. The main algorithm then operates on each 512-bit message block in turn, each block modifying the state. The 128-bit (16-byte) MD5 hashes (also termed message digests) are typically represented as a sequence of 32 hexadecimal digits. For example, a 43-byte ASCII input and the corresponding MD5 hash; MD5 ("The quick brown fox jumps over the lazy dog") = e4d909c290d0fb1ca068ffaddf22cbd0. Even a small change in the message will (with overwhelming probability) result in a completely different hash, due to the avalanche effect. For example, adding a period to the end of the sentence; MD5 ("The quick brown fox jumps over the lazy dog**.**”) =e4d909c290d0fb1ca068ffaddf22cbd0. Even a small change in the message will (with overwhelming probability) result in a completely different hash, due to the avalanche effect. For example, adding a period to the end of the sentence, MD5 ("The quick brown fox jumps over the lazy dog**.**") = e4d909c290d0fb1ca068ffaddf22cbd0. The hash of the zero-length string is MD5 ("") = d41d8cd98f00b204e9800998ecf84 27e, [9].

An interesting multi-letter cipher is the “Hill Cipher”, developed by the mathematician Lester Hill in 1929. The encryption algorithm takes m successive plaintext letter and substitutes for the m cipher text letters. The substitution is determined by m linear equations in which each character is assigned a numerical value (a = 0, b = 1 ... z = 25). For m = 3 the system can be described as follows. $C_1 = (k_{11}p_1 + k_{12}p_2 + k_{13}p_3)$ mod 26, $C_2 = (k_{21} P_1 + k_{12}p_1 + k_{23}p_3)$ mod 26, $C_3 = (k_{31}p_1 + k_{32}p_2 + k_{33}p_3)$ mod 26. This can be expressed in term of column vectors and metrics:

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k23 \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} \bmod 26$$

Or, C = kp mod 26, Where C and p are column vectors of length 3, representing the plaintext and cipher text, and k is a 3×3 matrix representing the encryption key. Operations are performed for mod 26, For example, consider the plaintext **PAY MORE MONEY** and use the encryption key

$$K = \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$$

The first three letters **PAY** of the plaintext are represented by the vector

$$\begin{pmatrix} 15 \\ 0 \\ 24 \end{pmatrix} \begin{pmatrix} 15 \\ 0 \\ 24 \end{pmatrix} = \begin{pmatrix} 375 \\ 819 \\ 486 \end{pmatrix} \bmod 26 = \begin{pmatrix} 11 \\ 13 \\ 18 \end{pmatrix} = \text{LNS}$$

Continuing in this fashion, the cipher text for the entire plaintext PAY MORE MONEY is LNS HDL EWM TRW, i.e. $C = E_K (P)$ = LNS HDL EWM TRW. Decryption requires using the inverse of the matrix K. The inverse K^{-1} of a matrix K is defined by the equation, $K K^{-1} = K^{-1}K = I$, where I is the identity matrix that is all zeros except for ones along the main diagonal from upper left to lower right. The inverse of a matrix does not always exist, but when it satisfies the preceding equation. In this case the inverse is:

$$K^{-1} = \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix}$$

This is demonstrated as follows:

$$\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} = \begin{pmatrix} 443 & 442 & 442 \\ 858 & 495 & 780 \\ 49 & 52 & 365 \end{pmatrix}$$

$$\bmod 26 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

It is easily seen that if the matrix K^{-1} is applied to the cipher text, then the plaintext is recovered [1][2][5][9][14][15][16].

The inverse of a square matrix A, sometimes called a reciprocal matrix, is a matrix A^{-1} such that $AA^{-1} = I$. Where, **I** is the identity matrix. Courant and Hilbert use the notation $\check{\mathbf{A}}$ to denote the inverse matrix. A square matrix **A** has an inverse if and only if the determinant $|A|\neq 0$. A matrix possessing an inverse is called non-singular, or invertible. The matrix inverse of a square matrix **m** may be taken in Mathematica using the function Inverse[**m**]. For a 2×2 matrix

$$A \equiv \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

The matrix inverse is

$$A^{-1} = \frac{1}{|A|} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

$$= \frac{1}{ab - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

3. CONVENTIONAL MESSAGE TRANSACTION APPROACHES

Presently, there are a number of approaches for electronic message transactions and most of them are based on message authentication codes (MAC). In message authentication operations, the following four important aspects are to be verified: (a) the contents of the messages have not been altered, (b) the source is authentic, (c) the message is timely, i.e., it has not been artificially delayed and replayed and (d) the sequence relative to other messages flowing between two parties is correct. In [1], four message transaction approaches have been presented based on MAC, Message authentication and confidentiality.

In the first approach, a secret-key cryptographic system (C) is applied on the message (M) with a secret-key (K) to produce the message authentication code (MAC) and then the produced MAC is concatenated with the message and send it to the receiver. In the receiving end, the receiver compute the MAC from the message by using the same cryptosystem and the same secret key and then compare it with the received MAC. If the generated MAC and the received MAC are equal, then it assures the authentication of the communicated message. The second approach provides the authentication but not the confidentiality, because the message as a whole is transmitted in the plaintext. Confidentiality can be provided by performing message encryption, and is done in the third approach with the MAC algorithm. In both the cases two separate keys are needed each of which is shared by the sender and the receiver. In the first case, the MAC is generated with the message as input and is then concatenated (| |) to the message. The entire block is then encrypted. In the second case, the message is encrypted first, then the MAC is generated using the resulting cipher text and is concatenated (| |) to the cipher text form the transmitted block.

A high level message authentication and confidentiality process has been presented in the fourth approach. In this process, two encryption techniques with two separate keys are used to encrypted message and encrypted MAC and concatenated them with each other and then send them to the destination. In the receiver end, MAC is generated and compared it two times to assure the authentication of the message with another level of security and then retrieve the message with confidentiality of the electronic transaction system. The studied approaches are used in different aspects with the imposing security services. Their security measures are presented in the following Table 1, [1][9].

4. METHODOLOGY

Our proposed secured electronic transaction technique with its methodology, algorithmic approach and the flow diagram of the system is derived in the following subsections.

Table 1: Comparative analysis for different message transactions

Approaches	Authentication	Confidentiality	Level of Security
Approach 1	√	X	None
Approach 2	√	√	Better
Approach 3	√	√	More Better
Approach 4	√	√	More More Better

4.1 Methodology for Message Transaction

The Methodology derives the steps to use the secure electronic message transaction using the proper digital signature scheme.

i. Preparing for Transaction: Initially we select the Transmitter.jar from the Runnable folder inside the Hill Cipher folder, which brings the window to write message and send to the receiver. At first, the sender has to write the User ID and Password, which is used to secure the message transaction. There is a big text box below, which is for the message to type and send to the receiver.

ii. Message Acceptance: The transacting message contains maximum size of 1024 kb.

iii. Destination Information: For sending messages to the receiver, the sender should put the IP address and the Port address. The Port address is set to 9000 by default to the both end. The IP address must be known by the sender for communicating with the receiver.

iv. Key Generation: A key length is taken to generate a key for the sender and for the receiver too. A key generator is created to generate the key which specifies the dimension of the square matrix which in this case is the encrypted matrix.

v. Digest Generation: For message digest, the method used is, the MD5 (Hash) Algorithm.

vi. Encryption Process: The number that is specified as the size of the matrix is the number of characters taken and it continues till the string is exhausted. Then it is putted in a column matrix and multiplied with the encrypted matrix to get the encrypted text. After the completion of the encryption process, the original message is then digested and concatenated (| |) with the encrypted message.

vii. Sending the Generated and Encrypted Key: After successful completion of the encryption, the sender hits the send button with declaring the correct destination address of the receivers and sends the transacted message.

viii. Decryption Process: Upon receiving the encrypted message, the receiver first separate the digested message from the encrypted message. Then decrypt the encrypted message and then calculate with the digest.

ix. Authentication Testing: If the received digests are matched, then the receiver can view the Authenticated message, otherwise an error will show during receiving the messages to the receiver.

4.2 Algorithmic Approaches

The algorithm has been divided into two phases. In the 1st phase, Sender Generates the message to send through some steps of encryption and in the 2nd phase the receiver receives the message and decrypt with some methods. The authentication process is done in the receiver end, if the authentication is done properly then the message is displayed or else it is damaged or dropped. The complete algorithm is given below.

Phase - 1: (Sender Site)
Step 1: Generating the Inverse Matrix (M^{-1}), so that (M X M^{-1}) is the IDENTITY MATRIX (I). Then send M^{-1} to the Receiver.

Step 2: Generating a Key, using the length of the Key Matrix (K) generating a Square Matrix (M).

Step 3: Taking the Message (N) and encrypting using the Key Matrix (M) following the "HILL CIPHER" to produce cipher text.

Step 4: Digesting (D) message (N), using MD5 and Concatenate || with the Cipher text (C), then sent to the Receiver.

Phase - 2: (Receiver Site)
Step 1: Received (C) is Decrypted using the M^{-1} (Key) received.

Step 2: Message Retrieved using the HILL CIPHER Decryption Technique.

Step 3: Digested using MD5

Step 4: Comparing the Received (D) by the Digest (D´).

Step 5: Comparison Succeeded, then the Message (M) is displayed in the output.

4.3 Flow Diagrams

A flow diagram is a graphical means of presenting, describing, or analyzing a process or the nature of the project. The boxes represent the steps through which the encryption process worked and established communication [3]. There are two flow diagrams shown in Fig 1 bellows.

A secured message transaction process has been designed, developed and implemented by using Java programming language. The contributions of section are to present the methodology of the system, algorithmic approach and the flow diagrams both in the sender site and the receiver site. This establishes the clear presentation of the proposed electronic message transaction system.

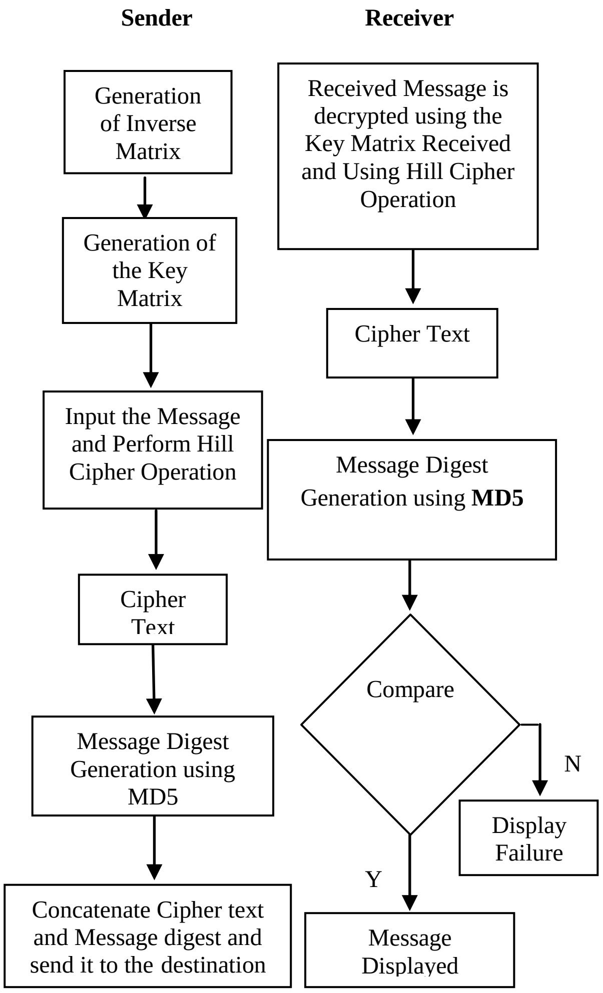

Sender
Receiver
Generation of Inverse Matrix
Generation of the Key Matrix
Input the Message and Perform Hill Cipher Operation
Cipher Text
Message Digest Generation using MD5
Concatenate Cipher text and Message digest and send it to the destination
Received Message is decrypted using the Key Matrix Received and Using Hill Cipher Operation
Cipher Text
Message Digest Generation using MD5
Compare
N
Display Failure
Y
Message Displayed

Fig 1: Flow diagram of the proposed system

5. IMPLEMENTATION

The proposed electronic message transaction system has been implemented using java programming. The possible errors of the system are identified and discussed. The software is executed for three examples and the produced cipher text and the retrieved message are analyzed and discussed in the following sub sections. For convenience, we just provided one test result.

5.1 Implemented Message Transaction Procedures

The message transaction software is built using the Java platform and digital signature scheme. The steps that have being implemented for secure message transaction have been given below
Step 1: The Runnable folder in the Hill Cipher contains four files:
1. Transmitter.jar = Transmission window to send message to the destination.
2. Receiver.jar = Receive message/check for message.
3. Key_Generator.jar = Generates Key and build a key file (key.mds).

4. Key.mds = It is a key file, which is manually placed to the folders of Transmitter, Receiver and Runnable.

Step 2: The Transmitter.jar contains ID, Password, Transaction Message, Encrypt Message, IP address, Port, Send, and Close.

Step 3: The Sender should complete the full input in the Transmitter.jar window. Any objects kept empty will display an empty error message while sending.

Step 4: The Sender inputs the ID and Password to ensure the authentication for transmission and the ID is being displayed in the receiver end, which is easier for the receiver to identify the sender. The encryption button is to see the message encryption codes. The IP address contains the destination or the receivers IP address and the Port address is assigned 9000 by default.

Step 5: After completing every input, then the send button is used to send the message to the receiver.

Step 6: The Receiver.jar file should be opened in the receiver end during the message transaction. If the sender succeeds to send message then the receiver end will have a message displayed in the Receiving window.

Step 7: The Receiver.jar contains a decrypt button to see the message received from the sender

5.2 Possible Errors

The possible errors of the developed system are presented below:

i. If the sender missed to input any field in the Transmitting window, then an input error or "Please enter password/id/message" will display.

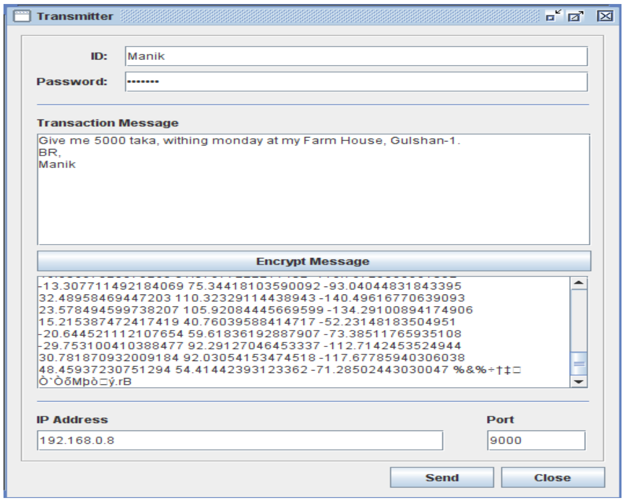
ii. If an invalid IP address is put then, no message will be send.

iii. If the key file doesn't matches with the destination key files, then key mismatch failure notice will display.

5.3 Analysis of Implemented Message Transaction System

The software that is build using the java language is tested below with some examples to proof the secure message transactions that are being done

Example: Manik is trying to communicate with the IP 192.168.0.8 writing some secret message, which will be encrypted and then will be finally sent to the destination IP using the port 9000. The sender Manik uses the Transmitter.jar window to write the message with inputting the following objects properly and sent to the receiver. The Encrypt Message displays the secret message, which is being encrypted by using the Hill Cipher encryption method.



ID: Manik

Password: ••••••••

Transaction Message

Give me 5000 taka, withing monday at my Farm House, Gulshan-1.
BR,
Manik

```
-13.307711492184069 75.34418103590092 -93.04044831843395
32.48958469447203 110.32329114438943 -140.49616770639093
23.578494599738207 105.92084445669599 -134.29100894174906
15.215387472417419 40.76039588414717 -52.23148183504951
-20.644521112107654 59.61836192887907 -73.38511765935108
-29.753100410388477 92.29127046453337 -112.7142453524944
30.781870932009184 92.03054153474518 -117.67785940306038
48.45937230751294 54.41442393123362 -71.28502443030047 %&%÷†‡□
Ò`ÒõMþò□ý.rB
```

IP Address: 192.168.0.8

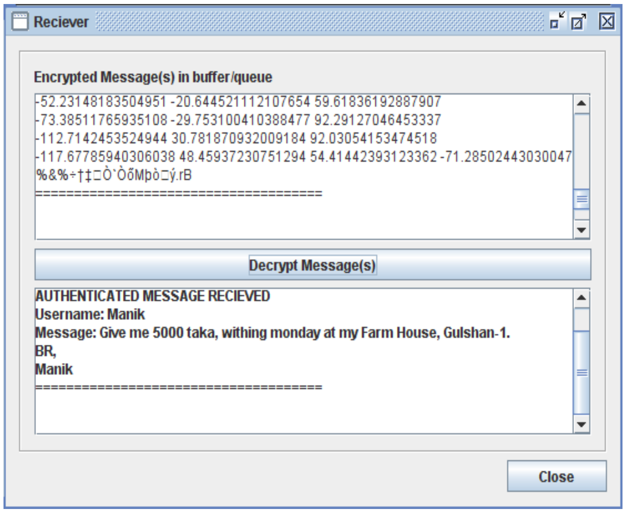
Port: 9000

Fig 2: Transmitting message to the receiver

Encrypted output:

```
9.66497263628157696.98158490011035-122.23055913189356
40.7899587909983485.39862649353896-110.0933698049997
20.10401821721097880.21020043880614-101.7981956271435
36.2209678532117583.62044316851348-107.6608234064239
30.77055780228806796.78513236781785-123.56074377213692
42.14486471536696492.05615122074849-118.37756286468192
20.10401821721097880.21020043880614-101.7981956271435
8.589559107266723110.01420649184068-138.0526401896412
91.5677406084256525.067564212237897-38.570280165597595
86.929899953940122.087811026709353-34.55013022897909
17.29262842158098646.0301395141484-58.9090790691742-
45.05667884634655124.88547278890984-152.40003559797807
35.16475688179930684.89293286213069-109.0786967046446-
49.37195668393571126.77560184004504-154.45149255777298
38.9113775640503190.73913351008736-116.69990657545358
36.4070728412081992.53623516615343-118.69253250341546-
7.76428274896709177.4171277269158-96.86516083789105
38.0656096909964492.5437086511831-118.70293974319966
6.83679240621565128.6963112254454-160.69938007247862-
4.755847033103109139.89874799007669-173.81872450085868
27.12293032657595499.76613702311434-126.15566768785042
29.94694594225448794.40410020876669-120.61409796770656-
27.84074892904488595.10676352874461-117.0426340494044
46.8065732337926991.97577222211402-118.76725333981092-
13.30771149218406975.34418103590092-93.04044831843395
32.48958469447203110.32329114438943-140.49616770639093
23.578494599738207105.92084445669599-134.29100894174906
15.21538747241741940.76039588414717-52.23148183504951-
20.64452111210765459.61836192887907-73.38511765935108-
29.75310041038847792.29127046453337-112.7142453524944
30.78187093200918492.03054153474518-117.67785940306038
48.4593723075129454.41442393123362-71.28502443030047
```

The above values are the encrypted message, which are taken to proof as the messages are sent securely to the sender or not.



```
-52.23148183504951 -20.644521112107654 59.61836192887907
-73.38511765935108 -29.753100410388477 92.29127046453337
-112.7142453524944 30.781870932009184 92.03054153474518
-117.67785940306038 48.45937230751294 54.41442393123362 -71.28502443030047
%&%÷†‡□Ò`ÒőMþò□ý.rB
=====================================
```

```
AUTHENTICATED MESSAGE RECIEVED
Username: Manik
Message: Give me 5000 taka, withing monday at my Farm House, Gulshan-1.
BR,
Manik
=====================================
```

Fig 3: Received message from the buffer

5.4 Result Analysis

From the above figures and codes that is being generated, it is determined that, a secure message communication is being occurred between the receiver and the sender. From the example we seen, *Manik* is trying to communicate with the receiver (192.168.0.8). Before this, to examine whether the message is securely transmitted or not, we compare the encrypted code before sending and after sending to the receiver. Generally it is proofed that, the message is securely transmitted to the destined receiver. The Input/output analysis makes a conclusion, which, for a digital electronic transaction of secure messages is proofed with this software, by verifying the codes and equalizing them. i.e.; secure message transaction is successful within the sender and the receiver. Any cryptographic system is measured when it works properly. The proposed electronic message transaction system has been implemented using java programming language. The java is selected as the programming language because of its simplicity, widely acceptance and better security.

6. SECURITY ANALYSIS

This section presents the fundamental security services that are established by our proposal, electronic message transaction system. Any security system establishes the fundamental security services those includes authentication, data confidentiality, Data integrity and non-repudiation. Our proposed system performs all the services properly. All the security services are analyzed and depicted bellows.

i. Authentication: Since the entire message in the communication system is encrypted with the dynamic Hill Cipher technique and performs the message digest operation in the sender site; and in the receiver site, the digest is again computed and compared with the received digest. So it establishes the authentication of the system.

ii. Data Confidentiality: The encryption technique of the proposed system is Dynamic Hill Cipher and it establishes the message confidentiality of the system. Encrypted message cannot known to all these except the genuine receiver who knows the key matrix.

iii. Data Integrity: Since the message of the transmitted system remain same and the modified message digest cannot matches with the received message. So this system establishes data integrity of the message in the proposed system

iv. Non-repudiation: Since only the sender and the receiver can know the key matrix and performs its inverse matrix, so the sender can not deny the communication and that is why, the system establishes the non-repudiation.

All the security services of a secured transaction system exist in the proposed system; and so on it is treated as a secured message transaction system. The established security services by the proposed electronic message transaction system are presented in the Table 2 bellows.

Table 2: Security services performed by the proposed system

System	Authentication	Confidentiality	Integrity	Non-repudiation
Proposed System	√	√	√	√

7. ACKNOWLWDGEMENT

We express our special thanks, sincere gratitude and deepest appreciation to Dr. M. Ismail Jabiullah, Associate Professor, Head, Department of Computer Science and Engineering, Institute of Science and Technology for his excellent guidance, kind supervision and continued encouragement to organize the materials diligently. During the process of working on this paper, he was a constant source of ideas, motivation, and encouragement. His valuable time, inspiration and scholarly advice helped us to accomplish the work systematically.

8. CONCLUSIONS

Secured message transactions are very much demandable in high-speed telecommunication networks. Several conventional message transaction approaches have been studied and realized. A secured electronic message transaction system has been designed, developed and implemented where Hill Cipher Cryptosystem is used as the encryption-decryption process with dynamic keys. For this process, any transacted message is to be inputted into the proposed system, and then a matrix of the message length is calculated with its inverse matrix. The calculated inverse matrix is sent to the receiver where he/she used it as the key for the cryptosystem. The key matrix and the Hill Cipher technique have been used to generate the cipher text. The message digest algorithm MD5 operations have been used on the message to generate the digest of the message. The generated digest is concatenated with the encrypted message and is sent to the destination. In the receiver end, the intended receiver first computes the digest of the message by performing the message digest algorithm MD5 and compares it with the received digest that establishes the data integration and non-repudiation of the proposed system. The receiver then decrypts the received encrypted message using the Hill Cipher technique with the received inverse matrix and retrieve the message properly which establishes the message confidentiality and authentication. The proposed system has been implemented using the Java programming and analyzed for several applications. Finally, the fundamental security services have been measured and analyzed. This may be applicable for cryptographic applications.

9. FUTURE WORKS

Research in secured electronic transactions system a diverse and mathematically sophisticated practice and so there are many scopes to do further works. In the current era of electronic communication, security issues are the top and central concern and thus are in high demand. The work in this paper implies that more effort must be spent in designing a security protocol for high demanding wireless communication for short time transactions or very effective in video conferences, and so group key generation techniques and distribution techniques will be needed for further works.

Key distribution process is a vital issue to build a secured electronic transaction system and so more effort should be put for designing more secured key distribution protocols and for key management protocols to design even better protocols. Security services are much more desirable for a secured electronic transaction system and so further effort may be applied to enhance the services including non-repudiation that is not considered in this study. Furthermore, to speed-up the electronic information transfers through the internet all the operations for key distribution, and security services should be reduced and be improved. It is possible to obtain faster, reliable, authenticate and non-repudiated service issues in the electronic transaction system. Now it is the plan to concentrate more efforts in this are in the future

10. REFERENCES

[1] Andrew S. Tanenbaum, Computer Networks, 4th edition, Prentice-Hall, Inc 2003, ISBN- 81-203-2175-8, pp. 731-732, 749-755,433-437

[2] Cryptographic Algorithms: http://www.eskimo.com/~weidai/algorithms.html

[3] Cryptography-RSA: http://www.cs.princeton.edu/introcs/79crypto/

[4] RSA Key Generator for default keys used: http://crypto.cs.mcgill.ca/~crepeau /RSA/generator_frame.html

[5] The Hill Cipher, http://www.math.sunysb.edu/~scott/papers/MSTP/crypto/8Hill_Cipher.html

[6] Shahrokh Saeednia, How to Make the Hill Cipher Secure, Cryptologia, 24(4), October 2000, pp353–360.

[7] "Hill Cipher Deciphered" provides an excellent explanation of computing matrix inverses with regard to the Hill cipher.

[8] An Introduction to Cryptology Prentice-Hall, ISBN 0-13-030369-0web services

[9] William Stallings, Cryptography and Network Security Principles and Practices, Fourth Edition

[10] Java Cryptography Architecture: Available at http://www.tucows.apollo.lv/javacorner/jdk1.1.6/docs/guide/security/CryptoSpec.html

[11] Dr. Y Lee, RSA Algorithm.pdf, Telecommunication forum, UK 2001, pp.1-5

[12] James Martin, Telecommunication And The Computer,3 ed, Prentice-Hall, Englewood Cliffs, N.J., 2001, pp.29-135,137-142

[13] RSA Laboratories, http://www.rsasecurity.com/node.asp?id=1012

[14] On the Key of the Hill Cipher, http://jeff.over.bz/papers/undergrad/on-the-keyspace-of-the-hill-cipher.pdf

[15] Dobbertin, Hans (1996). "The Status of MD5 after a recent attack". *CryptoBytes* **2**.

[16] “Hill Cipher Deciphered" provides an excellent explanation of computing matrix inverses with regard to the Hill cipher.