International Journal of Engineering Science, Advanced Computing and Bio-Technology
Vol. 9, No. 4, October – December 2018, pp. 138 - 145

Information Security using Matrix Transformation

T.N. Janakiraman[1], Nagaraj Thenkarai Janakiraman[2] and *Vishwashankar Thenkarai Janakiraman[3]

[1]*Department of Mathematics, National Institute of Technology, Tiruchirappalli-620015, Tamil Nadu, India, janaki@nitt.edu*

[2]*Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843, USA, tjnagaraj@tamu.edu*

*[3]*Department of Computer Science, Amrita Vishwa Vidyapeetham, Ettimadai, Coimbatore-641112, Tamil Nadu, India, tjv.shankar@gmail.com*

Abstract: *In this paper, sharing of information between two users is done with the help of matrix transformation. Messages are shared using three hand shake methodology with symmetric and asymmetric procedures. Also, two different procedures are given to implement the above sharing with good security and privacy among the users by embedding the messages, using matrix transformation.*

Key Words: *Information Security, Matrix Transformation, Symmetric and Asymmetric Keys, Key Sharing.*

1. Introduction

In practice, to share information between two or more users with expected privacy and security becomes an inevitable in information system. Here, the privacy and security force the system to make the procedure, in general to the intruders, with non-deterministic time complexity to decode the information shared among the users. In general, such crypto models either directly or indirectly use the property of prime numbers. Many schemes have been derived out of which, very few are practically used. In this paper, instead of using two prime numbers as used in the general practice, we proposed a methodology to use a transformation (using matrix with relatively prime numbers) to achieve good security in sharing of information. Some crypto system models are contributed by some researchers using special matrix transformations such as lower/upper triangular matrices [1] and Hilbert matrices [2] and [3]. In the proposed matrix transformation, we use multiple keys, which are either square matrices or vectors (column and row) with all their elements as relatively prime numbers. This methodology can also be further extended to share information between a group of multiple users.

In this paper, we give our schemes in three phases to implement our procedure in three different sections. In the first following section, we give a transformation of messages from person Alice and Bob. In second section, we give two procedures such that one is symmetric and another is hybrid symmetric. In the third section an asymmetric

Received: 09 October, 2018; Revised: 22 November, 2018; Accepted: 28 November, 2018
*Corresponding Author
DOI: 10.26674/ijesacbt/2018/49417

procedure is given. In the fourth section for secured transformation using sequences of embedded messages two methodologies are given. In the last section concluding remark is given.

Section 1

Scheme to share information between the Alice(P_1) and Bob(P_2):

In this section, we give a symmetric scheme in two phases to explain the sharing of information between two users. We also use only square matrices and row or column vectors in the proposed scheme for the keys. In addition, two primary keys (square matrices) and a secondary key (column vector) are allowed for each user with their elements, which are relatively prime in their nature.

Section 1.1

In this section, we present our scheme to pass information between two users namely Alice(P_1) and Bob(P_2) and to collect information by the Alice from Bob. First we assume Alice sends a message M_1 N_1K_1= A1=$(a_1,a_2,\ldots,a_n)^T$, where $K_1=(l_1,l_2,\ldots,l_n)^T$ to Bob using two keys M_1 and N_1 are n×n square matrices(two primary keys of Alice) and K_1(secondary key of Alice) and A_1 (public to Alice and Bob) are n-dimensional column vectors with the elements of M_1,N_1 and K_1 are relatively primes.

Let M_2 and N_2 be two primary keys, which are n×n square matrices and $K_2 = (m_1, m_2,\ldots,m_n)^T$ be secondary key of Bob with the similar nature of keys of Alice. Here, in turn, let us assume Bob sends message in two steps to Alice.

In the first step, let us assume that Bob computes $K_2^*=(a_1m_1,a_2m_2,\ldots,a_nm_n)^T$, $A_1^*=(a_1s_1,a_2s_2,\ldots,a_ns_n)^T$ and sends message $B_1^*=A_1^*\ (K_2^*)^T$ to Alice, where s_1, s_2,…,s_n are some dummy values used by Bob in the computation of A_1^*. Clearly, both Alice and Bob know the information about A_1 and B_1^*. From the above, it is clear that B_1^* is n×n square matrix with the property that each of its i^{th} column vector has GCD 'a_im_i' and each of its i^{th} row has GCD 'a_is_i' for i=1,2,…,n. Thus, Alice gets information about the key K_2 used by Bob, because Alice knows the information of 'a_im_i' and 'a_i'.

In step 2, First Bob transforms M_2 to M_2^* using some suitable transformation and sends $M_2^*\ (B_1^*-\lambda_{AB} I_n) = C_1$ to Alice for some suitable value of λ_{AB} known to both Alice and Bob, where I_n (identity n×n matrix) square matrix. The λ_{AB} value can be chosen as a combination of a_i, s_i and m_i for i=1,2,…,n, so that both Alice and Bob know about common value of λ_{AB} fixed by them. It also must be noted that the value of λ_{AB} should not be eigen value of B_1^*. Also, Alice and Bob know the information about B_1^* and C_1. From this, it is clear that Alice gets M_2^* (one of the primary keys of Bob) by computing $C_1(B_1^*-\lambda_{AB} I_n)^{-1}$ from the message sent by Bob, as $C_A^* = (B_1^*-\lambda_{AB} I_n)$ is invertible.

Thus, Alice gets information about the secondary key K_2 , M_2* (transformed key of M_2) sent by Bob but not the individual details of the keys M_2 and N_2.

Section 1.2

In this section, we give below the steps involved to share the key K_1 sent by Alice to Bob, which is similar to the steps given in 1.1.

Step-1: Bob sends M_2 $N_2K_2=B_2$ to Alice, where $B_2 = (b_1,b_2,\ldots,b_n)^T$, $K_2 = (m_1,m_2,\ldots,m_n)^T$(secondary key vector of Bob) and M_2 and N_2 are two primary keys of Bob with all its elements relatively prime to each other).

Step-2: In the first step, let us assume that Alice generates $K_1^*=(b_1l_1,b_2l_2,\ldots,b_nl_n)^T$, $B_2^*=(b_1t_1,b_2t_2,\ldots,b_nt_n)^T$ and sends message $A_1^* = B_2^*(K_1^*)^T$ to Bob. From the above, it is clear that A_1* is a n×n square matrix (Singular Matrix) with the property that each of its i^{th} column vector has GCD 'b_i l_i' for I = 1,2,…,n. From this Bob gets the key K_1 (the secondary key of Alice).

Step-3: Let M_1* be a transformed matrix of M_1 computed by Alice. Next Alice sends $M_1^*(A_1^*-\lambda_{AB}^*I_n) = C_2$ to Bob for some common value λ_{AB}* known to both Alice and Bob. Here, λ_{AB}* value can be chosen as a combination of b_i, l_i , t_i and m_i for i=1,2,…,n, so that both Alice and Bob know about common value of λ_{AB} fixed by them. It also must be noted that the value of λ_{AB} should not be eigen value of A_1*. Also, Alice and Bob know the information about A_1* and C_2. Now Bob computes $C_2(A_1^*- \lambda_{AB}^*I_n)^{-1} = M_1^*$, because Bob knows about $C_B^*= A_1^*-\lambda_{AB}^*I_n$, which is invertible. Thus, Bob obtains the secondary key K_1 and M_1*(transformed one of two primary keys of Alice).

Remark 1

1. Both Alice and Bob share their instant one of their primary keys (here M_1 and M_2) and their secondary keys (here K_1 and K_2).
2. Using which, they generate C_A* and C_B*, which are known to both and hence they are public keys.
3. As C_A*and C_B*are non-singular, they can generate a new invertible public key C_{AB}*common to both such that CA* $C_B^*= C_{AB}^*$ and use this new public key for passing messages between them. Also, if they ignore all previous keys used, in subsequent sharing of information, then that procedure is symmetric and also has some multiple securities.
4. Besides, if Alice and Bob use their remaining primary keys, which are not shared (here, N_1 and N_2) for decryption and the public keys M_1*, M_2* and C_{AB}* to encrypt the messages, then that procedure is an asymmetric.

Section 2

In this section, we give three procedures to share/pass further messages between Alice and Bob. In the first sub section, we give one symmetric procedure to use a single public derived key for both encryption and decryption of messages and another hybrid symmetric procedure, which provide multiple securities of message transformation. In the second sub section, we give third procedure, which is asymmetric to use both public and individual private keys to share messages between them.

Section 2.1

In this sub-section let us see the steps through which Bob gets message from Alice using single public key $C_{AB}^*$for both encryption and decryption of messages and hence it symmetric.

Now we give below the steps through which Alice passes messages to Bob.

Step-1: Alice sends $M_i\ C_{AB}^* = Q_i$ as a i^{th} message to Bob, where M_i is a new message sent by Alice encoding it as Q_i.

Step-2: As Bob has information about C_{AB}^*, Bob first computes $Q_i\ (C_{AB}^*)^{-1}$ and obtains the new message M_i sent by Alice.

Thus, Bob gets i^{th} message M_i sent by Alice for i = 1,2,….

As C_{AB}^* is known to Bob, Bob also can use the same methodology to send new message N_i to Alice.

Thus, this symmetric method is used to share messages between Alice and Bob.

Section 2.2

In this sub-section, we give hybrid symmetric procedure, which provides some additional security to pass message between Alice and Bob.

Step-1: First Alice sends $P_A F_A(C_{AB}^*) = A^*$ (derived public key) to Bob, where P_A is a private key of Alice and $F_A(C_{AB}^*)$ is a polynomial function on C_{AB}^* computed by Alice.

Step-2: Let $S_B A^*(C_{AB}^*)^{-1} = S_B^*$ be the encoded message of S_B, which sent by Bob to Alice.

Clearly, $S_B^* = S_B(P_A\ F_A(C_{AB}^*)(C_{AB}^*)^{-1})$. As Alice has information about C_{AB}^*, P_A and $F_B(C_{AB}^*)$, she computes $S_B^*(P_A F_A(C_{AB}^*)(C_{AB}^*)^{-1})^{-1}$ and obtains the new message S_B sent by Bob.

Thus, Alice gets message S_B sent by Bob with some additional security with the hybrid symmetric procedure, Here, even though C_{AB}^* is public key, Bob encodes his new message S_B first using the product of $A^*\ (C_{AB}^*)^{-1}$ and send it to Alice. Here, the private keys P_A and $F_A(C_{AB}^*)$ are known to Alice alone, but the product of them is known to be public for Alice and Bob.

Similarly, Bob can use a private keys P_B and her own polynomial function $F_B(C_{AB}^*)$ on C_{AB}^* to get a derived public key $P_BF_B(C_{AB}^*) = B^*$ and can get new messages from Alice. Thus, this hybrid symmetric/asymmetric method is used to share messages between Alice and Bob with multiple securities.

Section 3

In this section, let us see the steps through which Alice gets message from Bob. Here, first Alice sends a cipher text using public keys C_{AB}^*, M_1 and her private key N_1 with some additional private $F_A(C_{AB}^*)$ key to Bob. Then Bob using that cipher text and public keys M_1 and C_{AB}^*, encrypts a message and sends to Alice. Now, Alice using her private key N_1 and public key C_{AB}^*, decrypts the message. Thus, the following method is sharing is asymmetric.

Now, we give below the steps through which Alice gets messages from Bob.

Step-1: First Alice sends $M_1N_1F_A(C_{AB}^*)N_1M_1= A^*$ (derived public key) to Alice, where N_1 is a private key of Alice and $F_A(C_{AB}^*)$ is a polynomial function on C_{AB}^* computed by Alice.

Step-2: Let $S_1(M_1)^{-1}A^*(M_1)^{-1} = S_1^*$ be the encrypted message of secret message S_1 sent by Bob to Alice.

Clearly, $S_1^* = S_1N_1F(C_{AB}^*)N_{1.}$ As Alice has information about $F(C_{AB}^*)$ and $N_{1,}$ first she computes $S_1^*(N_1F_A(C_{AB}^*)N_1)^{-1}$ and obtains the secret message S_1 sent by Bob.

Thus, Alice gets message from Bob.

Following are the steps that how Bob gets messages from Alice using her private key N_2 , public keys C_{AB}^*, M_2 and his own polynomial function $F_B(C_{AB}^*)$.

Step-1: Let $M_2N_2F_B(C_{AB}^*)N_2M_2= B^*$ (derived public key) send by Bob to Alice, where N_2 is a private key of Bob and $F_B(C_{AB}^*)$ is a polynomial function on C_{AB}^* computed by Bob as an additional key for security.

Step-2: Let $S_2(M_2)^{-1}B^*(M_2)^{-1} = S_2^*$ be the encrypted message of secret message S_2 sent by Bob to Alice.

Clearly, $S_2^* = S_2N_2F_B(C_{AB}^*)N_{2.}$ As Alice has information about $F_B(C_{AB}^*)$ and $N_{1,}$ first she computes $S_2^*(N_2F_B(C_{AB}^*)N_2)^{-1}$ and obtains the secret message S_2 sent by Alice.

Thus, Bob gets message S_2 from Bob.

Thus, this asymmetric method is used to share messages between Alice and Bob.

Section 4

In this section, we shall see the steps to pass sequence of messages from Bob to Alice using the primary and secondary keys of them.

First in the following subsection let us see that how messages are embedded as sum of the messages in a sequence. Further we see the methodology that how they are encoded and decoded to retrieve new information.

Section 4.1

In this section, a sequence of information already passed are summed up to form an embedded message from Bob to Alice in the following steps using single public key.

Let PR(i) be defined as $\sum_{j=1}^{i} Mj$ (sum of all of all previous M_1 , M_2 , …. M_i messages), the embedded message that can be sent and retrieved similarly.

Step-1: Let PR(i)(SCF) = Q_i^* be an encoded message sent by Alice to Bob at i^{th} instant, where SCF is one of the scheme functions already used in the previous sections to encrypt the message to be sent from Alice to Bob.

Step-2: Now Bob decodes PR(i) using the given corresponding scheme function to decrypt the message used in the previous sections.

Step-3: Let M_{i+1} be a new message to be passed from Alice to Bob

Let PR(i+1)(SCF) = $Q_{i+!}^*$ be new encoded message sent by Alice to Bob. Now Bob can decode PR(i+1) and using PR(i), she computes M_{i+1} .

Thus, Alice and Bob share the information using a methodology of embedding of messages.

Remark 2

1. In this above section, we have given a procedure to embed the messages and share them between the users. In that, if we define PR(i) = $\prod_{j=1}^{i} Mj$.(That is the product of the matrices M_1, M_3, …, M_i) and use the similar steps given in section 4.1 to messages between the users then also we get appreciable security.
2. If some of the Matrix messages M_k are singular, they can be converted to non singular using the fact that (M_i - $\lambda_{AB}I_n$) is non singular for some common value of λ_{AB} defined in terms of the keys K_1, K_2, A_1^* and B_1^*. After passing them, they can be converted to original matrices afterwards as λ_{AB} is known.
3. All secret messages may be either n×n or m×n matrices.
4. Using the above procedure no intruder can easily tap the messages and hence this methodology provides high security of information, which is to be shared between Alice and Bob..
5. The above Sharing of Primary keys, secondary keys and messages between two persons can be extended similarly among n-users where n>=3 by posting all

primary keys K_1, K_2, K_3, …, K_n as column vectors of some primary square matrix n×n.

Conclusion:

In this paper, passing of information between two users are carried out with two primary (n×n matrices with relatively prime elements) and secondary keys(with relatively prime column and row vectors) using one symmetric, one hybrid symmetric and one asymmetric procedures. Besides, procedures to pass (a) sequence of messages and (b) a sequence of embedded messages and to retrieve original messages are given to maintain high security and privacy.

References:

[1] T.N. Janakiraman, Nagaraj Thenkarai Janakiraman and Vishwashankar Thenkarai Janakiraman, "Three Hand Shake Methodology to Share Information between Administrator and Multiple Users using Matrix Transformation", IJESACBT 2018.

[2] T.N. Janakiraman, Nagaraj Thenkarai Janakiraman and Vishwashankar Thenkarai Janakiraman, "Secured Symmetric, Hybrid Symmetric and Asymmetric Message Passing using Skew Symmetric and Coupled Matrix Transformation", IJESACBT 2018.

[3] T.N. Janakiraman, Nagaraj Thenkarai Janakiraman and Vishwashankar Thenkarai Janakiraman, "Sharing of Messages between Administrator and Multiple Users and using Orthogonal Transformation", IJESACBT 2018.

[4] T.N. Janakiraman, Nagaraj Thenkarai Janakiraman and Vishwashankar Thenkarai Janakiraman, "Generation of Private and Public Keys using Recursive Transformation to Share the Information Between Two Users in a Network ", IJESACBT 2018.

[5] T.N. Janakiraman, Nagaraj Thenkarai Janakiraman and Vishwashankar Thenkarai Janakiraman, "Application of recursive transformation in generation of multiple private and public keys and sharing of information among administrator and multiple users in a network", IJESACBT 2018.

[6] T.N. Janakiraman, Nagaraj Thenkarai Janakiraman and Vishwashankar Thenkarai Janakiraman, "Message passing using matrix transformation with asymmetric El Gamal type methodology", IJESACBT 2018.

[7] R Alvarez, FM Martinez and JF Vicent A, "A New Public Key Cryptosystem based on Matrices", WSEAS Information Security and Privacy (2007) 36-39 2007.

[8] S. Suresh Babu, "A symmetric cryptographic model for authentication and confidentiality using Hilbert matrix", Ph.D thesis, Andhra University, Visakhapatnam, 2010.

[9] Ahmed Yehya Mahmoud, "Development of Matrix Cipher Modifications and Key Exchange Protocol", Ph.D thesis, Eastern Mediterranean University January 2012

[10] William Stallings and Lawrie Brown, "Computer Security: Principles and Practice", Pearson Education, 2008.

Authors' Profile:

Dr.T.N.Janakiraman was born in Thiruneveli, Tamilnadu, India in 1960. He received his B.Sc., M.Sc. and Ph.D. degrees in Mathematics from Medras University, India in 1981, 1983 and 1992 respectively. He did his Post Doctoral in research during the period 1992-1994 sponsored by NBHM-DAE India. He joined as a faculty in the Department of Mathematics at NIT Trichy in 1994. At present he is a Professor of Department of Mathematics in National Institute of Technology Trichy .Link to his Profile: https://www.nitt.edu/home/academics/departments/maths/faculty/professor/janaki/

Nagaraj Thenkarai Janakiraman was born in Kanchipuram, Tamil Nadu, India in 1992. He received his B.Tech. degree in Electronics and Communication Engineering in 2013 at Amrita Vishwa Vidyapeetham, Coimbatore, Tamil Nadu, India. At present he is Graduate Research Assistant at Texas A&M University, College Station, Texas, USA. He has done Data Science PhD Intern at NVIDIA and SIEMENS where he designed and implemented a Cybersecurity workflow to identify potential network anomalies using Graph Analytics and worked on the DARPA Spectrum Collaboration Challenge. Developed and implemented a DeepReinforcement Learning framework to learn the optimal spectrum allocation policy of the networks in 2018.
Link to his Profile: https://www.linkedin.com/in/tjnagaraj/

Vishwashankar Thenkarai Janakiraman was born in Chennai, Tamil Nadu, India in 1997. He is doing his final year B.Tech. in Computer Science and Engineering from Amrita Vishwa Vidyapeetham, Coimbatore, Tamil Nadu. He has done part time two years Research Internship under Dr.T.N.Janakiraman, National Institute of Technology and having experience to work on the problems related to crypto models, prime factorisation problem, few Graph operations and with some approximation algorithms and their applications. He has also done Machine Learning Intern at VoxEdu, California in 2018 and developed a machine learning model to help non-native English speakers.
Link to his Profile: https://www.linkedin.com/in/vishwashankar-tj-004528141/