International Journal of Engineering Science, Advanced Computing and Bio-Technology
Vol. 9, No. 4, October – December 2018, pp. 146 - 151

Three-Way Handshake Methodology to Share Information between Administrator and Multiple Users 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, 'three-way handshake methodology' is employed to authenticate sharing information between multiple different users and administrator. Also the methodologies to share the information using matrix transformation with symmetric, hybrid symmetric and asymmetric procedures are given to ensure high security.*

Key Words: *Information Security, Matrix Transformation, Symmetric and Asymmetric Keys, Key Sharing, Three-Way handshake, Administrator, Multiple Users.*

1. Introduction

To share information between two or more users, both public key and authentication are employed to maintain secrecy and privacy. In order to have more reliability to share information between two or more users , the complexity of decoding procedure of general cipher text for any intruder need to be nondeterministic. Factorization of any integer into prime factors is in general not identified even now whether it is polynomial or not, because there is no efficient polynomial time algorithm available to find factorization of randomly assigned integer. We use this fact and employ a scheme to use matrix transformation as public and private keys and also for authentication. In the following sections, we give our scheme and implementation of procedure in various situations.

Section 1:

In this section, we give our procedure to share information between users and administrator using matrix transformation in two subsections.

Received: 12 October, 2018; Revised: 26 November, 2018; Accepted: 03 December, 2018
*Corresponding Author
DOI: 10.26674/ijesacbt/2018/49418

Throughout, we use M_i to denote either $m \times n$ or $n \times n$ matrix, K_i, G_i and C_i to denote $n \times 1$ column vectors for i = 1,2,…,n. Here, the matrices M_i and K_i have elements, which are large relatively primes/prime integers for i = 1,2,…,n.

1.1. Generation of public keys between ADMIN and other users

In this subsection, we give the procedure to generate public keys to share the information between ADMIN and other users P_i for i = 1,2,…,n. Let M_i and K_i be the primary and secondary keys used by the users P_i for i = 1,2,…,n respectively.

Let M_i $K_i = C_i$ be the cipher text sent by i^{th} user P_i to the administrator ADMIN. Let G_1, G_2,…,G_n be n ADMIN keys, which are used to authenticate the users to share information. Let the cipher texts $C_i = (c_1^{\,i}, c_2^{\,i},\ldots, c_n^{\,i})^T$, which is public key between P_i and ADMIN for i = 1,2,…,n.

Let $G_i = (g_1^{\,i}, g_2^{\,i},\ldots, g_n^{\,i})^T$ for i = 1,2,…,n.

Let $G_i^* = (c_1^{\,i} g_1^{\,i}, c_2^{\,i} g_2^{\,i},\ldots, c_n^{\,i} g_n^{\,i})^T$ for i= 1,2,…,n.

Let $C_i^* = (c_1^{\,i} \alpha_1^{\,i}, c_2^{\,i} \alpha_2^{\,i},\ldots, c_n^{\,i} \alpha_n^{\,i})^T$ for i = 1,2,…,n, where $\alpha_j^{\,i}$ are some arbitrary constants and let $\sum_j \alpha_j^{\,i} = \alpha_i$ for i = 1,2,…, n. In fact, the $\alpha_j^{\,i}$ are chosen in such a way that $\sum_j \alpha_j^{\,i} = \alpha_i$, which is not an eigen value of $C_i^*(G_i^*)^T$, which is a $n \times n$ matrix, for i = 1,2,…iii,n.

Let $D_{i,1}^* = C_i^*(G_i^*)^T$ be the ADMIN authentication to the user P_i for i= 1.2,…, n. It can be easily verified $D_{i,1}^*$ is $n \times n$ matrix and with the property that any l^{th} column of it has GCD $(c_l^{\,i} g_l^{\,I})$ and any m^{th} row of it has GCD $(c_m^{\,i} g_m^{\,I})$ for l, m and i = 1,2, …,n.

Now each of the users P_i can get the details of G_i and α_i and hence he can compute $B_i^* = (D_{i,1}^* - \alpha_i I_n)$, where I_n is unit matrix. But for any other intruders, getting G_i and α_i is not easy as it involves factorization of all elements of C_i^* and G_i^*, which is not so easy in general and hence the authentication issued by ADMIN could be easily extracted by P_i well before than intruder gets information and also keys can be randomized.

1.2 Symmetric and hybrid symmetric schemes to share messages between ADMIN and other users

Let ADMIN next send the message A_i ($m \times n$ matrix with m need not be n) to P_i encoding it as $D_{i,2}^*$ as follows for i= 1,2, …, n.:

Let A_i $B_i^* = D_{i,2}^*$, for i = 1,2, …, n (The message sent by ADMIN to P_i). Here, B_i^* is generated such that α_i is not eigen value of $D_{i,1}^*$ and hence B_i^* is non-singular, which is known to both P_i and ADMIN for i = 1,2, …, n. With this, P_i gets A_i by computing $D_{i,2}^* (B_i^*)^{-1}$, for i = 1,2, …, n. Thus, the messages A_i sent by ADMIN are received by P_i for i= 1,2,…, n respectively. As the above procedure uses the public single key for encryption and decryption of messages shared between ADMIN and users by P_i for i= 1,2,…, n the above procedure implemented is symmetric.

Next, we give a procedure to implement a hybrid symmetric procedure to pass information between ADMIN and other users.

Now let us assume the user P_i sends a request $R_i = F_i(B_i^*)\ PK_i$, where PK_i is a private key and $F_i(B_i^*)$ is a polynomial function on B_i^* generated by P_i, whose degree will be at the most n-1.

In turn, let ADMIN sends a secret message $S_{ADMIN,i}$ (m × n matrix with m need not be n) encrypting as $S_{ADMIN,i}^* = S_{ADMIN,i}\ (B_i^*)^{-1}\ R_i$. Now the user P_i gets $S_{ADMIN,i}$ by computing $S_{ADMIN,i}^*\ ((B_i^*)^{-1}F_i(B_i^*)\ PK_i)^{-1}$. Here, ADMIN has no idea of individual private keys PK_i and $F_i(B_i^*)$ of P_i. But ADMIN has a clue of product of these keys.

Similarly, ADMIN can send a request $R_{ADMIN,i} = F_{ADMIN,\ i}(B_i^*)\ AK_i$ to P_i, where AK_i is a private key and $F_{ADMIN,\ i}(B_i^*)$ is a polynomial function on B_i^* generated by ADMIN, whose degree will be at the most n-1.

In turn, let P_i send a secret message $S_{i,\ ADMIN}$ (m × n matrix with m need not be n) encrypting as $S_{i,ADMIN}^* = (B_i^*)^{-1}\ R_{ADMIN,i}$. Now the ADMIN gets $S_{ADMIN,i}$ by computing the following expression, $S_{ADMIN,i}^*((B_i^*)^{-1}F_{ADMIN,i}(B_i^*)AK_i)^{-1}$. Here, similarly P_i has no idea of individual private keys AK_i and $F_{ADMIN,\ i}(B_i^*)$ of ADMIN, but has details of product of these keys.

Thus, the above procedure of sharing of information can be treated as hybrid symmetric scheme between ADMIN and the users P_i, for i= 1,2, …, n. The above procedure adds additional difficulties to any intruder to get information and hence the above sharing is better than the symmetric scheme.

Section 2:

In this section, we give an asymmetric procedure to share the information between ADMIN and the users P_i. In this procedure first P_i sends a request $R_{i,\ ADMIN}$ to ADMIN as a cipher text using public keys B_i^*, PK_i^*(transformed n × n matrix of a private key PK_i which is also a n × n matrix) with some additional private key $F_i(B_i^*)$ key. Then ADMIN using that cipher text sends a secret message $S_{ADMIN,\ i}$. After that P_i decrypts the message sent by ADMIN using his private keys PK_i and $F_i(B_i^*)$. As the private key of P_i is not known to ADMIN the following method of sharing is asymmetric.

Now, let us see the steps involved in sharing of information through which P_i gets messages from ADMIN.

Step 1: First P_i needs to share the transformed key PK_i^* using one of the procedures given in Section 1.2 using B_i^* with ADMIN.

Step 2: Next P_i sends a request $(PK_i^*)^{-1}\ PK_i\ F_i(B_i^*)\ PK_i\ (PK_i^*)^{-1} = R_{i,ADMIN}^*$ (derived public key) to ADMIN, where PK_i is a private key and $F_i(B_i^*)$ is a polynomial function on B_i^* generated by P_i.

Step 3: Let $S_{ADMIN,i}\ ((PK_i^*)\ R_{i,ADMIN}^*\ (PK_i^*)) = S_{ADMIN,i}^*$ be the encrypted message of secret message $S_{ADMIN,i}$ (m × n matrix with m need not be n) sent by ADMIN to P_i.

Clearly, $S_{ADMIN,i}^{*} = S_{ADMIN,i}\, PK_i\, F_i(B_i^{*})\, PK_i$. As P_i has information about $F_i(B_i^{*})$ and PK_i, P_i computes $S_{ADMIN,i}^{*}(PK_i\, F_i(B_i^{*})\, PK_i)^{-1}$ and obtains the secret message $S_{ADMIN,i}$ sent by ADMIN. Thus, P_i decrypts the message sent by ADMIN using his private keys PK_i and $F_i(B_i^{*})$.

Now we give the reverse procedure that ADMIN gets message with the similar steps from P_i.

Here, ADMIN uses a private keys $PK_{ADMIN,i}$ and $F_{ADMIN,i}(B_i^{*})$ and public $PK_{ADMIN,i}^{*}$(transformed key of $PK_{ADMIN,I}$ a $n \times n$ matrix) gets messages from P_i.

Step 1: Here also, first ADMIN needs to share the transformed key $PK_{ADMIN,i}^{*}$ using one of the procedures given in Section 1.2 (using B_i^{*}) with P_i.

Step 2: Secondly, ADMIN makes a request $(PK_{ADMIN,i}^{*})^{-1}\, PK_{ADMIN,i}\, F_{ADMIN,i}(B_i^{*})\, PK_{ADMIN,i}\, (PK_{ADMIN,i}^{*})^{-1} = R_{ADMIN,i}^{*}$ (derived public key) to P_i, where $PK_{ADMIN,i}$ is a private key and $F_{ADMIN,i}(B_i^{*})$ is a polynomial function on B_i^{*} generated by ADMIN.

Step 3: Let $S_{i,\,ADMIN}\ ((PK_{ADMIN,i}^{*})\, R_{ADMIN,i}^{*}\, (PK_{ADMIN,i}^{*})) = S_{i,ADMIN}^{*}$ be the encrypted message of secret message $S_{i,ADMIN}$ ($m \times n$ matrix with m need not be n) sent by P_i to ADMIN.

Clearly, $S_{i,ADMIN}^{*} = S_{i,ADMIN}\, PK_{ADMIN,i}\, F_{ADMIN,i}(B_i^{*})\, PK_{ADMIN,i}$. As ADMIN has information about $F_{ADMIN,i}(B_i^{*})$ and $PK_{ADMIN,i}$, ADMIN computes $S_{i,ADMIN}^{*}(PK_{ADMIN,i}\, F_{ADMIN,i}(B_i^{*})\, PK_{ADMIN,i})^{-1}$ and obtains the secret message $S_{i,ADMIN}$ sent by P_i. Thus, ADMIN decrypts the message sent sent by P_i using his private keys $PK_{ADMIN,I}$ and $F_{ADMIN,i}(B_i^{*})$. Thus, this sharing of information between ADMIN and P_i for i= 1,2,..,n is asymmetric.

Section 3:

In this section, we give a procedure to share the information between ADMIN and Other multiple users by *S*ending sequence of messages one embedding on the previous messages as follows.

Let $PR_l(i) = \prod_{j=1}^{l} A_j^i$.(That is the product of the message matrices $A^i_1, A^i_2, \ldots, A^i_l$, which are already sent by ADMIN to P_i) for i =1,2, …,n.

If ADMIN wishes to send a new message A^i_{l+1} to P_i , then he can combine it with $PR_l(i)$ and compute $PR_{l+1}(i)$ and send it using any one of the procedures such as symmetric or hybrid symmetric or asymmetric discussed in section 1 and 2. Now, as P_i encrypts the message $Pr_{l+1}(i)$ and obtains that new message A^i_{l+1} sent by ADMIN.

Thus, the messages between ADMIN and P_i are shared, for i = 1,2, …, n.

Remark:

1. If some of the Matrix messages M are singular, they can be converted to non singular using the fact that $(M - \alpha_{ADMIN,\,P(i)} I_n)$ is non singular for some common value of $\alpha_{ADMIN,\,P(i)}$ defined in terms of the components of keys G_i and C_i , which are public to

ADMIN and P_i. After decryption, they can be converted to original matrices as $\alpha_{ADMIN,\ P(i)}$ is known to either side.

2. PR_l (i) can also be defined as $\sum_{j=1}^{l} A_j^i$ and embedded messages can be sent and retrieved similarly.
3. The admin messages A_i can be m × n matrix form for i = 1,2, …, n.
4. All keys are used with elements relatively prime element to use general terminology of coding theory for high security.
5. The messages sent by P_i to ADMIN at any instant can be embedded with the messages sent in the all previous instants to have very high security of message sharing.
6. Information between users P_i and P_j are passed through ADMIN in general.
7. If P_i and P_j want to share information between them, then they can make request to get common authentication $A_{i,j}$ to share information and share secret information $S_{i,j}$ among them using the procedures given such as symmetric or hybrid symmetric or asymmetric discussed in section 1 and 2.

Conclusion:

In this paper, sharing of information Between ADMIN and multiple users are implemented in symmetric, hybrid symmetric and asymmetric procedures. Besides, the embedding of messages and sharing messages are also given among to maintain very good security.

References:

[1] T.N. Janakiraman, Nagaraj Thenkarai Janakiraman and Vishwashankar Thenkarai Janakiraman, "Information Security using Matrix Transformation", International Journal of Engineering Science, Advanced Computing and Bio-Technology, Volume 9, No.4(October-December, 2018) pp 138-145. **DOI: 10.26674/ijesacbt/2018/49417.**

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

[8] 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/