Birendra Goswami, Dr.S.N.Singh / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 4, July-August 2012, pp.339-344

# Enhancing Security in Cloud computing using Public Key Cryptography with Matrices

[1]Birendra Goswami
Faculty Member, UMA, Ranchi

[2]Dr.S.N.Singh
HOD, Deptt. Of IT, XISS, Ranchi

**Abstract-**

**Cloud applications increasing demand for led to an ever growing need for security mechanisms. Cloud computing is a technique to leverage on distributed computing resources one do not own using internet facility in pay per use strategy on demand. A user can access cloud services as a utility service and begin to use them almost instantly. These features that make cloud computing so flexible with the fact that services are accessible any where any time lead to several potential risks. The most serious concerns are the possibility of lack of confidentiality, integrity and authentication among the cloud users and service providers. The key intent of this research work is to investigate the existing security schemes and to ensure data confidentiality, integrity and authentication. In our model symmetric and asymmetric cryptographic algorithms are adopted for the optimization of data security in cloud computing. These days encryption techniques which use large keys (RSA and other schemes based on exponentiation of integers) is seldom used for data encryption due to computational overhead. Their usage is restricted to transport of keys for symmetric key encryption and in signature schemes where data size is generally small. Public Key Cryptography with Matrices is a three-stage secured algorithm. We generate a system of non-homogeneous linear equations and using this system, we describe algorithms for key agreement and public encryption whose security is based on solving system of equations over the ring of integers which comes under the NP-Complete problems.**

*Keywords- **cryptography, encryption, decryption***

## I. Introduction

In the information age, cryptography has become one of the major methods for protection in all applications. Cryptography allows people to carry over the confidence found in the physical world to the electronic world. It allows people to do business electronically without worries of deceit and deception. In the distant past, cryptography was used to assure only secrecy. Wax seals, signatures, and other physical mechanisms were typically used to assure integrity of the message and authenticity of the sender. When people started doing business online and needed to transfer funds electronically, the applications of cryptography for integrity began to surpass its use for secrecy. Hundreds of thousands of people interact electronically every day, whether it is through e-mail, e-commerce (business conducted over the Internet), ATM machines, or cellular phones. The constant increase of information transmitted electronically has lead to an increased reliance on cryptography and authentication.

An obvious application of cryptography is the transformation of information to prevent other from observing its meaning. This is the classical concept of secrecy, wherein we attempt to prevent information from reaching an enemy in a usable form. Secrecy is viewed by many as the central issue in the field of information protection. Secure communication is the most straightforward use of cryptography. Two people may communicate securely by encrypting the messages sent between them.

The proposed algorithm for Public Key Cryptography with Matrices is a three-stage secured algorithm and it has a constant complexity (fixed number of multiplications) irrespective of the key size given over the ring of integers.

The algorithm can be divided into three stages functionally. The first stage involves the shuffling of the original data for which the linear congruential method is used and then the data is arranged in the form of a matrix of some dimension n x n.

In the second stage we traverse the matrix in a sequence of different forms like helical, spiral, etc after which the data is ready for the encryption process.

The third stage deals with generating the system of non-homogeneous linear equations from which we generate the private keys and it is highly impossible to solve this system of non-homogeneous equations over the ring of integers which guarantees the security of the data.

## 1. LITERATURE REVIEW

One basic research paper that helped in clarifying the concepts about basic Cryptography was

Introduction to Cryptography by Prof. Fred Piper of the Information Security Group of the University of London. Among other research papers that also helped in this respect are Asymmetric Cryptography and Practical Security by David Pointcheval and Contemporary Cryptology: Provable Security for Public Key Schemes also by David Pointcheval.

Another research paper by David Pointcheval and coauthored by Guillaume Poupard entitled A new NP-Complete Problem and Public Key Identification was found to be a major source of motivation for forming the new concepts.

A paper entitled Studying the Performance of Critical Neural Network on Problems related to Cryptography by E. C .Laskari, G. C. Meletios, D. K. Tasoulis and M. N. Vrchaits was also found to be very useful in this regard.

Other useful research papers include Neural Network and their Cryptographic Applications by David Pointcheval and Livreder and also the paper by Kyung-Ahshim and Song Sik Woo called Cryptanalysis of Tripartite and Multi-party Authenticated Key Agreement Protocols.

The implementation approaches for public key cryptography algorithms with matrices was mainly clarified by the following research papers.

The first paper was named Matrix based Asymmetric Bulk Encryption Algorithm and was written by Mukesh Kumar Singh of Texas Instruments, Inc. and another research paper that was found to be very helpful from the point of view of implementation was The Public Key Cryptography with Matrices also written by Mukesh Kumar Singh.

The other paper was by Farshid Delgosha and Farcmarz Fekri and is entitled Public Key Cryptography using Paraunity Matrices.

## 2. PROPOSED APPROACH FOR PUBLIC KEY CRYPTOGRAPHY WITH MATRICES

In this study an algorithm for public key cryptography using matrices will be proposed, which is structurally and functionally divided into two basic parts.

The first part deals with the pre-processing of data and it includes the two main processes of data shuffling and traversing of the data.

Finally there is the second part of the algorithm which deals with the key generation, key agreement and encryption /decryption processes.

Now we discuss the algorithms in greater details to explain its working and features:

### Section 1: PRE PROCESSING OF DATA.

This section will deal with the part of the algorithm that deals with the pre-processing of data and this section can be divided into the following two stages:

### 1.1) Shuffling of the Data

In this algorithm the idea for the shuffling of data is mainly based on that suggested by Mukesh Kumar Singh in his research paper *Matrix based Asymmetric Bulk Encryption Algorithm.*

Suppose L be the length of the message to be encrypted. We consider here two arrays as follows

1) index[1…….L] is an array containing all the indices of the message, and

2) hash[1……..M] is the array containing some magic numbers

such that when we apply the linear congruential method to the array index[1…L] then the output of the index[1…L] array does not contain any repeated indices.

The linear congruential method can be formulated as follows:

```
STEP 1) For I = 1 to  initialise index[I] = I, and initialise x = 1

STEP 2) For J = 1 to M, repeat Step (3) to Step (5)

STEP 3) For K = 1 to L, repeat Step (4) to Step (5)

STEP 4) x = ((J+1)*x + hash[(J+K) mod M]) mod L

STEP 5) Swap (index[K], index[x])

STEP 6) Return index[1…..L]
```

The original message is shuffled or rearranged on the basis of the array index[1….L] that we have obtained as an output using the method depicted above.

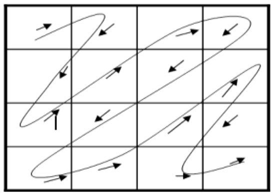
Now an integer N is selected such that $N^2 \geq L$, i.e. the value of $N^2$ is greater than or equal to that of L and the data is arranged in an NxN matrix form.

### 1.2) Traversing the Data Matrix

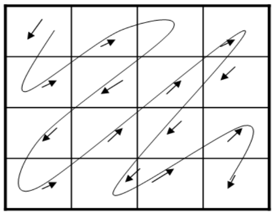
This stage involves reading out of the data from the data matrix of order NxN.

This can be achieved in any of the following manners which are depicted through appropriate self-explanatory diagrams:

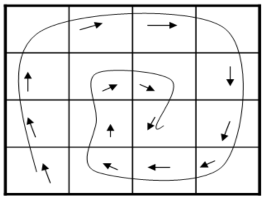
1.2.1) Spiral Traversal:



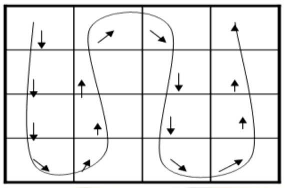
1.2.2)Reversed Spiral Traversal



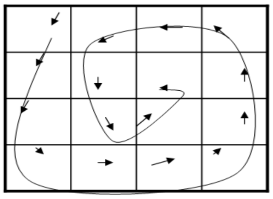
1.2.3) Helical Traversal



1.2.4) Sine Waveform Traversal



1.2.5) Reverse Helical



Since here there are five patterns, so there can be 5! possible sequences. Let the sequences be $T_1$, $T_2$, ……, $T_{120}$.*Suppose that the sequence represented above is $T_1$*
.

Next we discuss the three parts of the second stage of our algorithm:

**Section 2.1) KEY GENERATION**

We first take a matrix G of size nXn such that |G| = 0 and a list L = { $a_1$, $a_2$, …… ,$a_n$} of integers.
So, from section (2.1), we can form a circulant matrix
$L_c$ = circ($x_1$, $x_2$, ….. , $x_n$)
where the $x_i$'s are nothing but $a_i$'s.

Now, let σ be a permutation on the set {1, 2, 3, ……. , n}.
The matrix G can now be written in the form

$$G = \begin{bmatrix} R_1 \\ R_2 \\ . \\ . \\ R_n \end{bmatrix}$$ where the $R_i$'s are the rows of the matrix G.

Now, we try to find a Y such that
$R_i.Y = L_c\ \sigma(i)$
i.e. $R_i.Y = L_c(1,i)$ where i = 1, 2, 3, …., n.

The above system of equations can be put in the form
G.Y = X

where $$X = \begin{bmatrix} L_C(1,1) \\ L_C(1,2) \\ . \\ . \\ L_C(1,n) \end{bmatrix}$$

The above system of equations should be consistent and as we have taken |G| = 0, so the system cannot have a unique solution.
We take one such Y and from the elements of Y, we can form a circulant matrix $Y_c$.
Now, we formulate a matrix $P = L_c.G.Y_c$

Then we take
Public Key: { G, $L_c$, P}
Private Key: {$Y_c$, σ}

**SECTION 2.2) KEY AGREEMENT**

1. G is the matrix known to both the communicating parties.
2. Party 1 selects a private key {$L_c$ , $Y_c$, $\sigma_1$} by using a list L as mentioned in section (3.1). Find $T_1$ and send to PARTY 2
3. Party 2 will also select a private key {$M_c$, $Z_c$, $\sigma_2$ } using list M. Find $T_2$ and send to PARTY 1.
4. PARTY 1 receive $T_2$ and calculates $S=T_2.Y_c$
5. PARTY 2 receive $T_1$ and calculates $S=T_1.M_c$

Illustration of above process
PARTY 1
PARTY 2
Select { $Y_c$, $\sigma_1$ }
find $T_1$ = G. $Y_c$
send {$T_1$} → Gets
{$T_1$}
select {
$Z_c$, $\sigma_2$}
find $T_2$ =
G. $Z_c$
gets { $T_2$} ← send
{$T_2$}

find $S = T_2.\ Y_c$ find $S = T_1.\ Z_c$

For both the parties the commutative property of the product of circulant matrices ensures that

$S = G.\ Y_c.\ Z_c$

where matrix S is the shared secret between the two parties

## SECTION 2.3: ENCRYPTION AND DECRYPTION PROCESSES

### 2.3.1) Encryption Algorithm

Let S be the data matrix to be encrypted. We generate two circulant matrices $X_1$ and $X_2$ .

Calculate

$D_1 = X_1.G.X_2$

and

$D_2 = \{(X_1.P.X_2) \oplus S\}$

where $\oplus$ is the bitwise XOR operator between the corresponding elements of the two operand matrices.

The set $\{D_1, D_2\}$ is the encrypted form of the data S.

### 2.3.2) The Decryption Algorithm

In the above stage we obtained the encrypted data $\{D_1, D_2\}$ of the data S.

Now we can obtain the original data S back from this encrypted data by using the secret key of the communicating party as follows:

$$\begin{aligned} L_c.D_1.\ Y_c \oplus D_2 &= L_c.X_1.G.X_2.\ Y_c \oplus X_1.P.X_2 \oplus S \\ &= L_c.X_1.G.X_2.\ Y_c \oplus X_1.L_c.G.\ Y_c.X_2 \oplus S \\ &= L_c.X_1.G.X_2.\ Y_c \oplus L_c.X_1.G.X_2.\ Y_c \oplus S \\ &= 0 \oplus S \\ &= S \end{aligned}$$

Hence we have decrypted the data using the private key of the communicating party.

## SECURITY AND ILLUSTRATION OF THE ALGORITHM ON PUBLIC KEY CRYPTOGRAPHY WITH MATRICES

In this chapter, the proposed algorithm will be explored from the perspective of security and then we illustrate the working of the algorithm with appropriate examples.

## SECURITY OF THE ALGORITHM

The possible attacks on the security of this algorithm can be either by directly solving the system of equations G.Y = X as in section (3.1) or using the matrix $P = L_c.G.Y_c$ .

To avoid the first possibility of attack, we have to select the augmented matrix [G:X] such that the rank of [G:X] = rank of G = r < n, where n is the number of unknowns. Then the n-r variables are independent and so we can take any arbitrary values for these variables and the remaining r variables are dependent on these n-r variables.

So, if we are able to find matrices G and X such that the value of rank r is minimum and that of n is more, then it will be ensured that the number of independent variables is high, so that searching the solution becomes an NP-Complete problem.

In the second case, the intruder can try to know the value of $Y_c$ using $P = L_c.G.Y_c$.

By using the list L given in the public key, he can have possibly n! values of $L_c$ and so the intruder have n! different systems of equations of the form $P = L_c.G.Y_c$ , which again gives rise to NP-Complete problem.

## ILLUSTRATION OF THE WORKING OF THE ALGORITHM

Now the encryption-decryption process is illustrated by using the sample data as below:

IT IS GENERALISED KEY

Here the total length of the data is L = 21.

**Data Shuffling:**

Using section (2.1), we shuffle the data using

Hash = {13, 91, 11, 12, 78, 37, 77, 17 }

Then the Index array values become

Index(1) = 12Index(2)=7 …………. Index(21) = 10

Now the data will be shuffled to the following form

AGYDE ENT SLIISEIKR E

Take A = 11, ……….. ,Z= 36 and take the space as 99.

As mentioned in section (2.1) , we will arrange the data in a matrix of size 5X5 to get

$$A=\begin{bmatrix} 65 & 71 & 89 & 68 & 69 \\ 32 & 69 & 78 & 84 & 32 \\ 83 & 76 & 74 & 74 & 83 \\ 69 & 73 & 75 & 82 & 32 \\ 69 & 0 & 0 & 0 & 0 \end{bmatrix}$$

**Data traversing:**

Using section (2.2), the matrix A can be traversed with $T_1$ .

Then we get the data matrix

$$S=\begin{bmatrix} 0 & 74 & 69 & 69 & 65 \\ 83 & 0 & 73 & 74 & 84 \\ 69 & 82 & 78 & 32 & 83 \\ 89 & 75 & 32 & 68 & 0 \\ 32 & 76 & 71 & 69 & 0 \end{bmatrix}$$

**Key Generation:**

Now the data matrix is encrypted using section (3).

Let public key be taken as

$$G = \begin{bmatrix} 10 & 9 & 18 & 20 & 2 \\ 9 & 30 & 44 & 28 & 19 \\ 20 & 18 & 36 & 40 & 4 \\ 18 & 60 & 88 & 56 & 38 \\ 40 & 36 & 72 & 80 & 8 \end{bmatrix}$$

and
L = { 160, 320, 78, 80, 39 }
Take
$L_c$ = circ( 78, 320, 18, 160, 39)

$$\text{and } \sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 5 & 4 & 1 & 2 \end{pmatrix}$$

Then we have to find a $Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix}$ such that G.Y = X as mentioned in section (3.1).

Then we get the equations

$$10y_1 + 9y_2 + 18y_3 + 20y_4 + 2y_5 = 80$$
$$9y_1 + 30y_2 + 44y_3 + 28y_4 + 19y_5 = 39$$
$$20y_1 + 18y_2 + 36y_3 + 40y_4 + 4y_5 = 160$$
$$18y_1 + 60y_2 + 18y_3 + 20y_4 + 38y_5 = 78$$
$$40y_1 + 36y_2 + 72y_3 + 80y_4 + 8y_5 = 320$$

Now, solving the above equations we get the values of the variables

$$y_1 = (683 – 48a – 116b + 37c )/73$$
$$y_2 = -10(33 + 278a/10 + 10b +172c/10)/219$$

where $y_3 = a$, $y_4 = b$ and $y_5 = c$ such that a,b,c are integres.

So, from the above context we can observe that the values of $y_3$ ,$y_4$ and $y_5$ can be any arbitrary integers and the values of $y_1$ and $y_2$ depends on these values.

Now, taking $y_1 = -6$, $y_2 = 5$, $y_3 = 10$, $y_4= 7$ and $y_5 =50$ and $Y_c$= circ(50 5 -6 10 7).

Therefore, the private key is {$Y_c$, σ}.

And P= $L_c$.G.$Y_c$

$$\begin{pmatrix} 906682 & 1612876 & 2128210 & 1517420 & 860248 \\ 1079353 & 1554188 & 2016671 & 1912644 & 615370 \\ 1099921 & 1888605 & 2485698 & 1860577 & 967603 \\ 1234786 & 1808536 & 2350222 & 2179208 & 738100 \\ 764882 & 1093930 & 1418596 & 1357554 & 427766 \end{pmatrix}$$

**Encryption:**

Take two circulant matrices
$X_1$ = circ( 19 89 72 29 28 )
$X_2$ = circ( 26 29 42 52 57 )
and then

$D_1 = X_1.G.X_2$

$$\begin{pmatrix} 1592898 & 1588391 & 1389547 & 1272655 & 1435519 \\ 1810771 & 1806917 & 1581634 & 1446233 & 1633173 \\ 2153039 & 2144688 & 1874571 & 1721060 & 1937992 \\ 1661041 & 1683017 & 1491604 & 1316741 & 1524393 \\ 1344113 & 1339086 & 1170567 & 1074362 & 1210054 \end{pmatrix}$$

$$D_2 = \{(X_1.P.X_2) \oplus S \}$$
$$= 1.0e+010 *$$
$$\begin{pmatrix} 7.8250 & 7.6111 & 6.8724 & 6.5195 & 7.2816 \\ 8.1174 & 7.8883 & 7.1075 & 6.7749 & 7.5465 \\ 7.4335 & 7.2391 & 6.5555 & 6.1788 & 6.9262 \\ 6.9448 & 6.7503 & 6.0853 & 5.7938 & 6.4579 \\ 7.6472 & 7.4302 & 6.6922 & 6.3845 & 7.1082 \end{pmatrix}$$

{$D_1$, $D_2$} is the encrypted data of the data matrix S.

**Decryption:**

$$L_c.D_1.Y_c \oplus D_2 = \begin{bmatrix} 0 & 74 & 69 & 69 & 65 \\ 83 & 0 & 73 & 74 & 84 \\ 69 & 82 & 78 & 32 & 83 \\ 89 & 75 & 32 & 68 & 0 \\ 32 & 76 & 71 & 69 & 0 \end{bmatrix} = S$$

## 3. Limitations

Cloud Computing is a way of providing dynamically scalable and available resources such as computation, storage etc as a service to users who can use it to deploy their applications and data. Cloud Computing can handle data in both the public and the private domain. But this seemingly harmless way of thinking about building applications has its own set of issues.

## 4. Conclusion

At the beginning of this study it was realized that in recent times key encryption including RSA and other schemes based on exponentiation of integers is rarely used for data encryption because of the larger overhead in terms of processor time utilization.Their use is more or less restricted to transport of keys for symmetric key encryption and in signature schemes where data size is generally small. Another observation that was made is that these days all symmetric/asymmetric encryption algorithms are using hash functions as an integral part for the message integrity.

To encrypt large messages a hybrid approach is used in which the messages are actually encrypted using symmetric schemes (DES, AES, etc.) and the key is transported using asymmetric schemes (RSA). In the algorithm that has been proposed here the effort has been in the direction of faster public key encryption without compromising the security of the system. The Public Key Cryptography with Matrices is a three-stage secured algorithm and it has a constant complexity (fixed number of multiplications) irrespective of the key size given over the ring of integers.

Public Key Cryptography with Matrices is a three-stage secured algorithm. We have generated a system of non-homogeneous linear equations and using this system, we have described the algorithm for key agreement and public encryption whose security is based on solving system of equations over the ring of integers which comes under the NP-Complete problems.

## II. REFERENCES

[1] I.N. Herstein, topics in Algebra 2nd Edition. willey Eastern Limited India Edition.

[2] Mukesh Kumar Singh, Matrix Based Asymmetric Bulk Encryption Algorithm, Proceedings of the 2004 IEEE Workshop on information Assurance United States Military Academy West point, NY 10-11 june

[3] W.Diffie and M. Hellman, “New Directions in Cryptography”, IEEE Transactions on Information theory.

[4] David I,Steinberg Computational Matrix Algebra International Student Edition Mc Graw-hill

[5] Daniel T. Finkbeinger,11 Introduction To Matrices And linear Trans- Formations Second Edition Freeman International Edition.

[6] michael D.Greenberg Advanced Engineering Mathematics Second Edition Low Price Edition.

[7] Toeplitz and Circulant Matrices: A review , Robert M.Gray ,Information Systems laboratory, Department of Electrical Engineering ,Stanford University.

[8] T.ElGamal, A Public Key Cryptosystem and a Signature Scheme based on Discrete Algorithms, IEEE Transactions on Information Theory,31:469-472 , July 1985.

[9] Kunth D.E .The Art Of Computer programming volme2/ Semi-numerical Algorithms. Addision Wesely Publishing company.

[10] The Original RSA Patent as filed with the U.S. Patent Office by Rivest; Ronald L. (Belmont, MA), Shamir; Adi (Cambridge, MA), Adleman; Leonard M. (Arlington, MA), December 14, 1977, Patent Number 4405829.

[11] [AES] Advanced Encryption Standard (AES), National Institute of Standards and Technology, FIPS PUB 197 (November 2001). Available at http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

[12] [Adam] Adams, Carlisle, Steve Lloyd: Understanding Public Key Infrastructure Concepts, Standards & Deployment, Macmillan Technical Publishing, Indianapolis, 1999.

[13] R. Rivest, A. Shamir, and L. Aldeman, “A method for obtaining digital signatures and public-key cryptosystems,” Commun. ACM, vol. 21, no. 2, pp. 120–126, 1978.

[14] A. K. Lenstra, “Unbelievable security. Matching AES security using public key systems,” in Advances in Cryptology—ASIACRYPT’01,ser. Lecture Notes in Computer Science, C. Boyd, Ed. New York: Springer, 2001, vol. 2248, pp. 67–86.

[15] D. Boneh and H. Shacham, “Fast variants of RSA,” CryptoBytes, vol.5, no. 1, pp. 1–9, 2002.

[16] M. J. Hinek, “Another look at small RSA exponents,” in Topics in Cryptology-CT-RSA 2006, ser. Lecture Notes in Computer Science, D.Pointcheval, Ed. New York: Springer, 2006, vol. 3860, pp. 82–98.

[17] G. Qiao and K.-Y. Lam, “RSA signature algorithm for microcontroller implementation,” in Smart Card Research and Applications,CARDIS’98, ser. Lecture Notes in Comput. Sci., J.-J. Quisquater and B. Schneier, Eds. New York: Springer, 1998, vol. 1820, pp. 353–356.

[18] H.-M. Sun and M.-E.Wu, An approach towards Rebalanced RSA-CRT with short public exponent Cryptology ePrint Archive, Report 2005/053, 2005 [Online]. Available: http://eprint.iacr.org/2005/053

[19] H.-M. Sun, M. J. Hinek, and M.-E. Wu, On the design of Rebalanced- RSA, revised version of [37] Centre for Applied Cryptographic Research, Technical Report CACR 2005-35, 2005 [Online]. Available: http://www.cacr.math.uwaterloo.ca/techreports/2005/cacr2005-35.pdf

[20] H.-M. Sun and C.-T.Yang, “RSA with balanced short exponents and its application to entity authentication,” in Public Key Cryptology—PKC 2005, Lecture Notes in Computer Science. NewYork: Springer, 2005, vol. 3386, pp. 199–215.

[21] H.-M. Sun, W.-C. Yang, and C.-S. Laih, “On the design of RSA with short secret exponent,” in Advances in Cryptology—ASIACRYPT’99, ser. Lecture Notes in Computer Science, K.-Y. Lam, E. Okamoto, and C. Xing, Eds. Berlin: Springer, 1999, vol. 1716, pp. 150–164