Article can be accessed online at http://www.publishingindia.com

# Fully Homomorphic Encryption with Matrix based Public Key Crypto Systems

**Addepalli VN Krishna*, Addepalli Hari Narayana**, Kokk Madhura Vani*****

**Abstract**

In this work, a novel mechanism is considered for asymmetric mode of encrypting data. A generator matrix is used to generate a field with a large prime number. The generator matrix, prime number and quaternary vector are used as global variables. Those global variables are used to calculate public key and also sub keys which in turn are used in the ElGamal mode of encryption. The decryption of data is done with Private Key. The proposed algorithm supports the features like authenticity of users, security & confidentiality of data transmitted. The mechanism can well be used in homomorphic encryption where computations are carried out on cipher text and generate an encrypted result which, when decrypted, matches the result of operations performed on the plaintext.

Going by the construction of the algorithm, encryption is being done on blocks of data for which it consumes less computing resources. Going by complexity of the algorithm, the key length needed is much less to provide sufficient strength against crypto analysis.

**Keywords:** Homomorphic Encryption, Public and Private Keys, Elgamal Mode, Crypto Analysis and Complexity.

## Introduction

In cryptography, a public key is a value provided by some designated authority as an encryption key that, combined with a private key derived from the public key, can be used to effectively encrypt messages and digital signatures.

Homomorphic encryption is a form of encryption which allows specific types of computations to be carried out on cipher text and generate an encrypted result which, when decrypted, matches the result of operations performed on the plaintext. There are several efficient, partially homomorphic cryptosystems, and a number of fully homomorphic, but less efficient cryptosystems. Although a cryptosystem which is unintentionally homomorphic can be subject to attacks on this basis, if treated carefully homomorphism can also be used to perform computations securely.

A somewhat homomorphic computation supports only one operation (either addition or multiplication) on plaintexts. A cryptosystem which supports both addition and multiplication (thereby preserving the ring structure of the plaintexts) is known as fully homomorphic encryption (FHE) and is far more powerful. Using such a scheme, any circuit can be homomorphically evaluated, effectively allowing the construction of programs which may be run on encryptions of their inputs to produce an encryption of their output. Since such a program never decrypts its input, it can be run by an untrusted party without revealing its inputs and internal state. The existence of an efficient and fully homomorphic cryptosystem would have great practical implications in the outsourcing of private computations, for instance, in the context of cloud computing.

## Literature Survey

Kahrobaei, Koupparis and Shpilrain (2009) talked about a matrix public key based on Deffie Hellman, but they used

* Professor & Head, CSE Department, Navodaya Institute of Tech. Raichur, Karnataka, India.
E-mail: hari_avn@rediffmail.com
** Student, 2nd Year, Electrical Engineering, Madhya Pradesh, India.
*** Assistant Professor, CSE Department, VITS, Hyderabad, Telangana, India.

(small) matrices over a group ring of a (small) symmetric group as the platform. This work is compared with DDH & CDH problems. Mahalanobis (2013) specified the importance of matrix keys in public key cryptography. Goswami (2013) worked on the importance of matrix key based public key encryption systems in cloud environment. Guo-Ping and Chun-Sheng (2013) studied the importance of Ergodic Matrices in public key based crypto systems. A crypto analysis on Ergodic based systems are studied and discussed by Chun-Sheng (2012). An elementary study is discussed in DSN progress report (2007) which studies public key systems based on Algebraic Coding theory. Krishna, Pandit and Babu (2007) and Krishna and Babu (2009) discussed symmetric and probabilistic encryption schemes based on matrix keys. The work also discusses the relevance of probabilistic encryption in the context chosen cipher text crypto analysis. In another work the author discusses the use of weak and strong matrix keys in encryption process in the context of crypto analysis. Craig (2009) discusses the concept of Homomorphic encryption is discussed in relevance to Ideal Lattices. The concept is discussed in terms of both somewhat homomorphic and fully homomorphic encryption schemes.

## Methodology of the Proposed Work

This work is divided into following modules

1. Converting Mathematical model to Cryptosystems
2. Identifying Global parameters
3. Identifying Public and Private Keys.
4. Generating Basins (Sub Key values)
5. Encryption and Decryption
6. Fully Homomorphic Encryption
7. Crypto analysis
8. Conclusion & Future work
9. References

## Converting Mathematical Model to Cryptosystems

A matrix generator is considered. For a large prime number a field is generated with the considered generator. The rank of the field is given by the number of matrix values formed in the field. By considering a random integer as a private key, Public key is calculated using the prime field and the generator. Known the Public key and Private Key, the model is ready for encryption process.

## Global Parameters

The generator matrix, prime number and quaternary vector form the global variables.

## Public and Private Keys

Let n be the private key. The generator matrix powered to the private key will form the public key matrix.

## Basins (Sub Key Values)

Known the public key matrix, it is powered with a random number. The output matrix is multiplied with quaternary vector. The output of this process is multiplied with sign function. A sequence is generated. Considering the similarity of values of this sequence, Basins are formed. These basins constitute different values in different basins. These basins form sub-keys to convert plaintext to cipher text.

## Encryption and Decryption

### Encryption

$P_B^r$ is multiplied with quaternary vector to generate a sequence which is divided into basins ($G_3$).

$$(P_m + \text{Basins}),\ G^r = (P_m + G_3),\ G_1 = C1, C2$$

### Decryption

$C_2^t$ is multiplied with quaternary vector to generate a sequence which is divided into basins ($G_3$).

$$P_m + G_3 - G_3 = P_m$$

## Somewhat Homomorphic Encryption

### Encryption

$$(P_{m1} + \text{Basins}),\ G^r = (P_{m1} + G_3),\ G_1 = C11, C12$$

$$(P_{m2} + \text{Basins}),\ G^r = (P_{m2} + G_3),\ G_1 = C21, C22$$

Where C12 = C22 = C2.

$$C1 + C2 = (P_{m1} + P_{m2}) + 2(\text{Basins}),\ 2G^r$$

### Decryption

$(2G^r)^{t*}$ Quaternary Vector = 2(Basins)

$P_{m1} + P_{m2}$ = C1+C2 – 2(Basins)

## Fully Homomorphic Encryption

### Encryption

$(P_{m1} + \text{Basins}), G^r = (P_{m1} + G_3), G_1 = C11, C12$

$(P_{m2} + \text{Basins}), G^r = (P_{m2} + G_3), G_1 = C21, C22$

where C12=C22

$C1*C2 = (P_{m1*}P_{m2}) + \text{Basins}^2 + \text{Basins}\ (P_{m1+}P_{m2}), G^{2r}$

### Decryption

$P_{m1} * P_{m2} = C1*C2 - \text{Basins}\ (C1+C2-2(\text{Basins})) - \text{Basins}^2$

$P_{m1} * P_{m2} = C1*C2 - \text{Basins}\ (C1+C2) + \text{Basins}^2$

**Example**

Random Generator matrix considered (g) = [12 34 11; 12 23 22; 32 34 33];

Field (P) = 37;

Considered Quaternary Vector of 64 values with base 4 ie (Q)

Private Key considered (n) =8

Public key generated (PB) = $g^8$ = [20 18 36 16 2 36 11 10 0];

**Encryption Process**

Random number considered for each block of data(r) = 4;

Length of Plaintext considered = 64 bits.

$(PB^{4)} * Q = (G3)$

$(PB^{4)}$ = [17 30 18; 18 26 30; 16 4 22];

Q = [0 0 0;0 0 1;0 0 2;0 0 3;0 1 0;….;3 3 0;3 3 1;3 3 2;3 3 3]

Sign (G3) =k

Converting k to integer form i.e. r (3, 1) +4*r (2, 1) +16*r (1, 1)

The sequence formed is

0 0 2 10 0 0 42 42 0 40 42 42 40 40 b42 42 0 0 10 42 0 21 42 42 32 42 42 42 40 42 42 42 0 2 42 42 0 42 42 42 40 42 42 42 42 42 42 42 2 42 44 42 42 42 42 42 42 42 42 42 42 42 42 42 42;

K= Basins formed by considering equality of values.

B(1) =[ 1,2,5,6,9,17,18,21,33,37];

B (2) = [3, 34, 49];

B (3) = [4, 19];

B(4)=[7,8,11,12,15,16,20,23,24,26,27,28,31,32,35,36,38,39,40,42,43,44,45,46,47,48,50,52,53,54,55,56,57,58,59,60,61,62,63,64];

B (5) = [10, 13, 14, 29, 41];

B (6) = [22];

B (7) = [25];

B (8) = [51];

C2= $g^4$ = [19 28 27 33 10 30 34 35 17];

Plaintext= welcome to 2015.

Converting to alphanumerical value

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 32 | 14 | 21 | 12 | 24 | 14 | 29 | 24 | 2 | 0 | 1 | 5 |
| Basins | 1 | 2 | 5 | 6 | 9 | 1 | 18 | 21 | 33 | 37 | 3 | 34 |
| Add | 33 | 16 | 26 | 18 | 33 | 31 | 47 | 45 | 3 | 37 | 4 | 39 |
| Mod 37 | 33 | 16 | 26 | 18 | 33 | 31 | 10 | 8 | 35 | 0 | 4 | 2 |
| CT(C1) | x | g | q | J | x | v | a | 8 | z | 0 | 4 | 2 |
| CT(C2) | K | t | s | x | a | u | y | z | i | | | |

**Decryption**

C2 is powered to private key (n) and multiplied with Q to generate G3.

Known C1 and G3 plain text can be retrieved.

## Crypto Analysis

1. A random number is used for each block of data during encryption process. This helps the data to be free from side channel attacks.
2. Variable length sub keys are used as basins which makes the process to be free from linear crypto analysis.
3. A sign function is used in both encryption and decryption process which makes the process to be free from differential crypto analysis.
4. Public key is developed on discrete logarithm problem which itself is a hard problem.

## Conclusion and Future Work

The work considers a public matrix key for encryption process which is based on discrete logarithm problem.

Since discrete logarithm problem is a hard problem it gives sufficient strength to the algorithm. A sign function is used to develop Basins which makes the process free from linear & differential crypto analysis. The random number used in the encryption process makes the process to be free from side channel attacks. The work is also applicable in fully homomorphic encryption process. The present work handles data encryption at block level of plain text. The work can also carried out for encryption of data at character level of plain text.

## References


Chun-Sheng, G. (2012). *Crypto analysis on Public key encryption scheme Using Ergodic matrices over GF(2)*. Advances in Technology and Management, Advances in Intelligent and Soft computing, 165, 129-135.

DSN Progress Report. (2007). A public key crypto system based on algebraic coding theory, 42-44,114.

Gentry, C. (2009). A Fully Homomorphic Encryption Scheme (Ph.D. thesis).

Gentry, C. (2009). *Fully homomorphic encryption using ideal lattices*. In 41st ACM Symposium on Theory of Computing *(STOC)*.

Goswami, B. (2013). Enhancing security in cloud computing using public key cryptography with matrices. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(8).

Guo-Ping, J., & Chun-Sheng, G. (2012). *A Novel Public Key Cryptosystem Based on Ergodic Matrices Over GF(2)*. International Conference Computer Science & Service System (CSSS).

Kahrobaei, D., Koupparis, C., & Shpilrain, V. (2013). *Public Key Exchange using Matrices over Group rings*. Lecture Notes, Groups Complexity Cryptology, 5(1), 97-115.

Krishna, A. V. N., & Pandit, S. N. N. (2004). A new algorithm in network security for data transmission. *Acharya Nagarjuna International Journal of Mathematics and Information Technology*, 1(2), 97-108.

Krishna, A. V. N. (2005). *A* simple algorithm for random number generation. *Journal for Scientific & Industrial Research*, October, 64, 794-796.

Krishna, A. V. N., Pandit, S. N. N., & Babu, A. V. (2007). A Generalized scheme for data encryption technique using a randomized matrix key. *Journal of Discrete Mathematical Sciences and Cryptography*, 10(1), 73-81.

Krishna, A. V. N., & Babu, A. V. (2009). Training of a new probabilistic encryption scheme using an optimal matrix key. *Georgian Electronic & Scientific Journal*, 2(19), 24-34.

Mahalanobis, A. (2013). Are matrices useful in Public key cryptography. *International Mathematics Forum*, 8(39), 1939-1953.

Stuntz, C. (2010). What is Homomorphic Encryption, and Why Should I Care?