International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682
Vol.5, No.9, September 2016 DOI:10.15693/ijaist/2016.v5i9.28-32

HOMOMORPHIC ENCRYPTION SCHEME USING HILL CIPHER FOR CLOUD DATA SECURITY

D.Chandravathi†
GVP College for Degree and PG courses(A),
Rushikonda, Viakhapatnam-45

Dr.P.V.Lakshmi††
GITAM University,
Rushikonda, Viakhapatnam-45

ABSTRACT-Cloud computing usage has increased rapidly in both industries and in research. It is a broad and diverse phenomenon where users are allowed to store large amount of data on cloud storage for futureuse. Data security, privacy, confidentiality, integrity and authentication are the various security issues which are to beaddressed. As the data is stored and processed away from the user and the cloud service provider, privacy and integrity of the data plays a crucial role.Most of the cloud service provider stores the data in plaintext format and user need to use their own encryptionalgorithm to secure their data if required. The data needs to be decrypted whenever it is to be processed. This paper aims at Homomorphic encryption using Hill cipher encryption algorithm for the security of data in cloud environment. Encrypted data is stored in cloud .The cipher key which is generated for encrypting the data plays a major role.It focuses onstoring data on the cloud in the encrypted format using additive homomorphic encryption.

Keywords: Data security, Hill cipher , homomorphic encryption, Cloud, additive homomorphic encryption.

I INTRODUCTION

In recent scenario the concept of public-key cryptography reconsiders is Homomorphic Encryption. This technology expanded with the concern of security of data in federated cloud networks[1][3]. It has been revealed that the security of data is a major hindrance to implement theservices in cloud. The need for innovative security models for user access to cloud resources is highly required as data is moved between disparate networks which allow companies or organizations to offload the data in a secured manner. Many algorithms have evolved based on homomorphic techniques the provides very strong security requirements for data in the past few years[4]. The proposed model aims to improve the security during data retrieval in cloud scenario without the need to use a centralized control over the encryption and decryption techniques.

Both these party shares their work on top of the encrypted data. The proposed model aims at performing arbitrary computations on the encrypted data called, homomorphic techniques, which give rise to privacy; the model tends to perform critical operation on encrypted data. Homomorphic encryption is evolved to solve such critical issues. Usingoperations over encrypted bits homomorphic encryption data protection is achieved through additive and multiplicative [5]. The cloud service provider, without being aware of its content accepts encrypted user query data to perform processing. The resultant cipher is generated which is sent to the user in the encrypted form. The user decrypts the data alone and views the result. The public-key and private-key cryptosystems are designed with various fault attacks.Construction of an encryption scheme that is both additively and multiplicatively homomorphic is a major challenge[4][5]. Partial homomorphic and Fully homomorphic are the major concern which uses additive and multiplicative homomorphism respectivelyas set of operations. The first cipher key generation takes place atthe user level using Hill cipher key algorithm.

Homomorphic Encryption

Encryption has traditionally been viewed as a mechanism that enables secure communication [4]. In particular, Public-key Encryption provides a way for Alice to encrypt a message into a ciphertext using Bob's public key, and for Bob to decrypt the ciphertext to obtain the message using his secret key. In this view of encryption schemes, access to encrypted data is all or nothing – having the secret decryption key enables one to learn the entire message, but without the decryption key, the ciphertext is completely useless [8].

Can we do arbitrary computations on data while it remains encrypted, without ever decrypting it? This state of affairs raises an intriguing question, first posed by Rivest, Adleman and Dertouzos in 1978, which promoted the idea of performing computations on encrypted data without being able to "see" the data. Such ability also gives rise to a number of useful applications including the ability to privately outsource arbitrary computations to the "cloud" and the ability to store all data encrypted and perform computations on encrypted data, decrypting only when necessary[1][2].

Fully Homomorphic encryption is a special type of encryption system that permits arbitrarily complex computation on encrypted data. Homomorphic encryption is the conversion of data into ciphertext that can be analyzed and worked with as if it were still in its original form. Homomorphic encryption is expected to play an important part in cloud computing, allowing companies to store encrypted data in a public cloud and take advantage of the cloud provider's analytic services.

Homomorphic encryption permits computing on encrypted data. That is, the client can encrypt his data x and send the encryption **Enc(*x*)** to the server. The server can then take the ciphertext **Enc(*x*)** and evaluate a function f on the underlying x obtaining the ***encrypted***result **Enc(*f*(*x*))** The client can decrypt this result achieving the wanted functionality, but the server learns nothing about the data that he computed on. Homomorphic encryption is **functional encryption**, where our goal is to reveal the result of the computation to the server, but protect all other information about our encrypted input.

For example, a user sends a request to add the numbers 1 and 2, which are encrypted to become the numbers 33 and 54, respectively. The server in the cloud processes the sum as 87, which is downloaded from the cloud and decrypted to the final answer, 3.A normal symmetric cipher -- DES, AES is not homomorphic. The RSA algorithm is homomorphic but only with respect to multiplication.

Homomorphic encryption schemes are malleable by design. **Malleability** is a property of some cryptographic algorithms[3]. An encryption algorithm is malleable if it is possible for an adversary to transform a ciphertext into another ciphertext which decrypts to a related plaintext.

II Related work

Homomorphic Encryption

In 1978, the notion of encryption schemes that permits nontrivial computation on encrypted data was first proposed by Rivest, Adleman and Dertouzos. Rivest proposed the exponentiation function and the RSA function as additive and multiplicative privacy homomorphisms, respectively. Note, however, that neither of these functions by themselves provides even chosen plaintext security[11].

There are several partially homomorphic cryptosystems, and also a number of fully homomorphic cryptosystems. Although a cryptosystem which is unintentionally malleable can be subject to attacks on this basis, if treated carefully homomorphism can also be used to perform computations securely.

The main difference between partial homomorphic cryptosystem and fully homomorphic cryptosystem is, in partial homomorphism padding of data will not be done while in fully homomorphic cryptosystems padding of data is implemented.

1. Partially homomorphic cryptosystems

- Unpadded RSA
- ElGamal
- Goldwasser–Micali
- Benaloh
- Paillier
- Other partially homomorphic cryptosystems

2. Fully homomorphic encryption

- Early homomorphic cryptosystems
- Gentry's cryptosystems
- Cryptosystem over the integers
- The 2nd generation of homomorphic cryptosystems

Cloud Computing

The information technology model for computing is composed of all the IT components like hardware, software, networking, and services. It is necessary to enable the development and delivery of cloud services via the Internet or a private network. Cloud Provider and Cloud User are the prominent actors in Cloud Computing. Cloud Provider is the enterprise that provides cloud services[9]. A Cloud Users are organizations, educational institutes, individuals utilizing the cloud services. Hence, there is a necessity for security, confidentiality and visibility with respect to the cloud providers. The main aspect is to protect the data from hacking.

Cloud Security

At present both in Public Cloud and Private Cloud, security should be provided to encrypt data that is stored and also to provide secure transmission from a local machine to a cloud data store. The stored data is encrypted and the channel of data transmission is well secured with key exchanges. But actually performing computations on the data stored in the cloud requires decrypting it first, which makes critical data available to the cloud provider. Data Mining and other Data Analysis onto the Encrypted Database is a far distant thing to achieve by using available encryption standards. The proposal here is to encrypt data before sending to the cloud providers. Thereby performing computations on clients' data at their request, such as analyzing sales patterns, without exposing the original data[10]. To achieve this it is also necessary to hold the cryptosystems based on Homomorphic Encryption either a Fully Homomorphic Encryption (FHE) or Somewhat Homomorphic Encryption (SHE).

III SYSTEM MODEL

The system framework for the proposed model is as shown in Figure-1. The framework explains about encryption process with encrypted data along with the keys that are shared between the clouds in the federated network[4]. The framework explains about the components involved and their functionalities. The user uploads the data in an encryptedform. The key generation for encryption technique is doneby using a random key for Hill cipher algorithm. The encrypted data is stored in the data server. Key Generation is performed with the encrypted data on the file using matrix cipher keygeneration algorithm where the plain text from the user istaken as input in the form of matrices. This model proposes a modified cipher key function F which introduces the novel obfuscations in the matrix along withthe key matrix and hence this cipher cannot be broken by the brute force attack which provides an additional strength to the cipher.

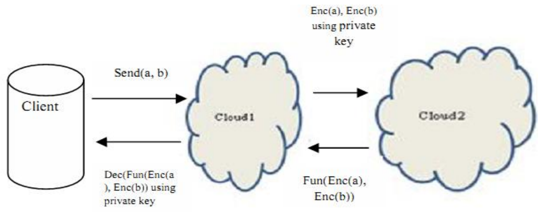
Enc(a), Enc(b) using private key
Send(a, b)
Client
Cloud1
Cloud2
Dec(Fun(Enc(a), Enc(b)) using private key
Fun(Enc(a), Enc(b))

Fig 1

Decryption is carried with the inverse of the key . Let m and c denote theplaintext/message and cipher text of the integer respectively. Here is an additive cipher that encrypts a block of four letters. Our plaintext messages split into blocks Numbers are substituted for letters by a = 1, b = 2, … , z = 26. The key adds to each component of the block – thought of as a column vector – component-wise. Decryption is accomplished by adding the additive inverse of the (vector) key to the ciphertext.

To encrypt a four-letter block, the key is a 4 X 1matrix. There are 4, 26, 456076 = possible keys – one of which produces plaintext. If we know one plaintext/ciphertext block correspondence, we can solve for the key. This additive cipher is not truly a block cipher because changing one plaintext letter of a plaintext block changes only one letter of the corresponding ciphertext block. A multiplicative cipher using matrices produces a true block cipher.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} \frac{d}{ad-ac} & \frac{-b}{ad-bc} \\ \frac{-c}{ad-bc} & \frac{a}{ad-bc} \end{bmatrix}$$

IV METHODOLOGY

Algorithm:

1. Encryption Algorithm

The steps of the encryption algorithm are as follows:

Step1: Start
Step 2: Input: Plaintext, Key Matrix.
Step 3: Convert the plaintext characters in to matrix form .
Step 4: Partitions input block into two halves mxn [(mxn/2) and (mxn/2)]
Step 5: Perform Encryption using C = KP MOD26 Where C & P are the matrices of order1 X N. Also K is a matrix of the order NXN.

2. Homomorphic encryption:

Step 1: Multiply with the left half data of inputmatrix with the key matrix.
Step 2: Similarly perform on right half of the dataof input matrix with key matrix to getcipher matrix.
Step 3: Perform permutation by swapping halves.Then the two halves pass through n roundsof processing then Combine Then the twohalves pass through n rounds of processing then combine to produce the cipher block.
Step 4: Each round i has as input Li-1 & Ri-1 derived from the previous round as well as a sub-key ki derived
Step 5: Computation is done for each round.

3. Decryption Algorithm

The steps of the Decryption algorithm are as follows:
Step1: Start
Step 2: Input : Ciphertext, Key matrix
Step 3: Calculate inverse key. If the determinant ofthe Key matrix is> zero Then set offset as follows:
If (Determinant > =0) Then set offset =1 Else Set offset = -1 .
Step 4: Decryption: P = CK-1 Mod 26.

Example:

Plaintext : gvpg

Converted into Matrix form:

$$\begin{bmatrix} g & v \\ p & g \end{bmatrix} = \begin{bmatrix} 6 & 21 \\ 15 & 6 \end{bmatrix}$$

MOD Operation:

Perform the MOD operation by using the value 26, will get the result

$$\boldsymbol{Key} = \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix}$$

1. Homomorphic Encryption:

Take the left half of the matrix

$$\begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix} * \begin{bmatrix} 6 \\ 15 \end{bmatrix} = \begin{bmatrix} 63 \\ 87 \end{bmatrix} \text{mod } 26$$

$$C1 = \begin{bmatrix} 11 \\ 9 \end{bmatrix}$$

$$\text{Right Half} = \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix} * \begin{bmatrix} 6 \\ 21 \end{bmatrix} = \begin{bmatrix} 81 \\ 72 \end{bmatrix} \text{Mod } 26$$

$$C2 = \begin{bmatrix} 3 \\ 20 \end{bmatrix}$$

Cipher Text is joining C1 and C2

$$C = \begin{bmatrix} 11 & 3 \\ 9 & 20 \end{bmatrix}$$

2. Homomorphic Decryption:

We will take the cipher text and key

Inverse = $K.K^{-1} = K^{-1}.K = I$

$$(ad - bc) * \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = 9^{-1} \begin{bmatrix} 5 & -3 \\ -2 & 3 \end{bmatrix}$$

$$= 9^{-1} \begin{bmatrix} 5 & -3 \\ -2 & 3 \end{bmatrix} * \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix} \text{Mod } 26$$

$$= 9^{-1} \begin{bmatrix} 5 & 23 \\ 24 & 3 \end{bmatrix}$$

- a^x =1 mod m
- x= a^{-1} mod m
- 9x=mod 26
- 9(3)=(1) mod 26
- 27=mod 26
- 27/26=1 (remainder)

$$= 3 * \begin{bmatrix} 5 & 23 \\ 24 & 3 \end{bmatrix} \text{Mod } 26$$

$$= \begin{bmatrix} 15 & 17 \\ 20 & 9 \end{bmatrix} * \begin{bmatrix} 11 & 3 \\ 9 & 20 \end{bmatrix}$$

$$= \begin{bmatrix} 15 & 17 \\ 20 & 9 \end{bmatrix} * \begin{bmatrix} 11 \\ 9 \end{bmatrix} = \begin{bmatrix} 318 \\ 301 \end{bmatrix} \text{Mod } 26$$

$$= \begin{bmatrix} 6 \\ 15 \end{bmatrix} \begin{bmatrix} 15 & 17 \\ 20 & 9 \end{bmatrix} * \begin{bmatrix} 3 \\ 20 \end{bmatrix} = \begin{bmatrix} 385 \\ 240 \end{bmatrix} \text{Mod } 26$$

$$= \begin{bmatrix} 21 \\ 6 \end{bmatrix} = \begin{bmatrix} 6 & 21 \\ 15 & 6 \end{bmatrix} = \begin{bmatrix} g & v \\ p & g \end{bmatrix}$$

Finally, we get the plaintext.

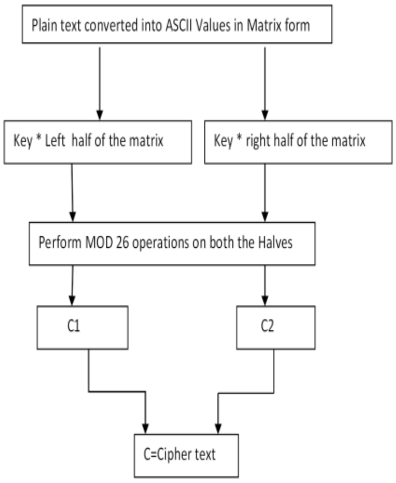
Plain text converted into ASCII Values in Matrix form
Key * Left half of the matrix
Key * right half of the matrix
Perform MOD 26 operations on both the Halves
C1
C2
C=Cipher text

Fig 2

The figure 1.2 describes Homomorphic encryption for plain text. The decryption is by taking the inverse of the matrix.

V Results

The proposed model is analyzed by executing setof experiments. The experimental results are as shown inFigure-3 and Figure-4. With respect to Table 1 and Table 2.

File size (KB)	Encryption Time (Msec)
13	0.23
28	0.28
32	0.37
45	0.47
49	0.49
56	0.54
67	0.59
80	0.77

Table 1

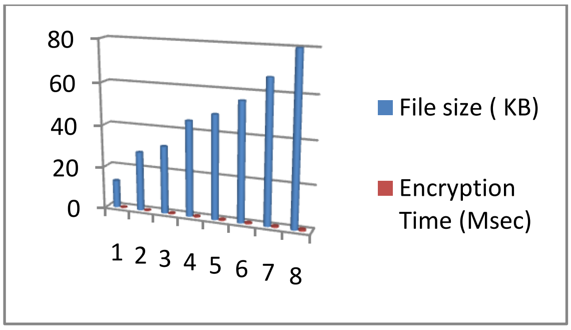

Fig 3

File size (KB)	Decryption Time (Msec)
13	0.43
28	0.52
32	0.56
45	0.67
49	0.74
56	0.98
67	1.24
80	1.45

Table 2

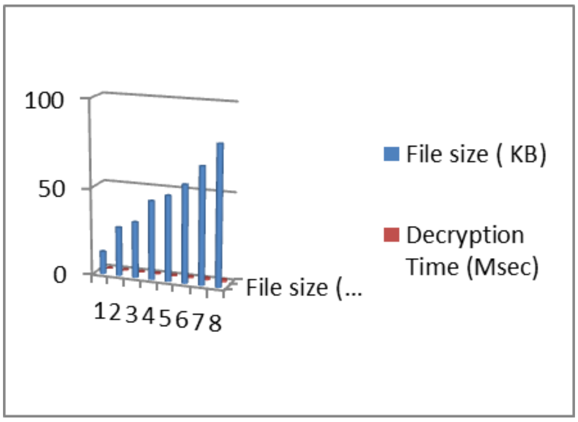

Fig 4

VI CONCLUSION

This paper motivates and solves the problem of data security in cloud environment using homomorphic encryption technique. The proposed homomorphic based encrypted technique preserves the data from invisibly leaking the sensitive information. The mechanism of the cloud devise a new technology which makes the data owner confident on the security of the data stored in cloud environment, since the encryption and decryption keys cannot be compromised. By security analysis, we show that the proposed scheme guarantees data privacy. In future, this existing system can be further improved by the use of variable length key and Unicode values.

VII References

1. Craig Gentry, A Fully Homomorphic Encryption Scheme,http://crypto.stanford.edu/craig/craig-thesis.pdf, 2009.
2. Understanding Homomorphic Encryption http://en.wikipedia.org/wiki/Homomorphic_encryption .Computing Blindfolded: New Developments in Fully Homomorphic Encryption VinodVaikuntanatham.
3. Homomorphic encryption based Data Security on Federated Cloud Computing by G *Anitha* and Vijayakumar V. Computer Science and Engineering, Sri Venkateswara College of Engineering, Pennalur, Sriperumbudur, India School of Computing Sciences and Engineering, VIT University, Kelambakkam, India.
4. C. Cachin, I. Keidar and A. Shraer, Trusting the cloud, ACM SIGACT News, Vol. 40, pp. 81-86, 2009.
5. J. Hendricks, G. R. Ganger and M. K. Reiter, Lowoverhead byzantine fault-tolerant storage, Proceedings oftwenty-first ACM SIGOPS symposium on Operating systems principles, ACM, pp. 73-86, 2007.
6. ChiragModi, Dhiren Patel, Bhavesh Borisaniya, Avi Patel, and Muttukrishnan Rajarajan, “A survey on security issues and solutions at different layers of Cloud computing”, Journal of Super Computers, pp.561–592,2013.
7. J. Ravi kumar and M. Revati, “Efficient Data Storage and Security in Cloud “, In Proc. International Journal of Emerging trends in Engineering and Development, vol.6, no.2, 2012.
8. Shizuka Kaneko, Toshiyuki Amagasa and Chiemi Watanabe, “Semi-Shuffled BF: Performance Improvement of a Privacy-Preserving Query Method for a DaaS Model using a Bloom filter”, in Proc. International Conference on Parallel and Distributed Processing Techniques and Applications, 2011.
9. Aguilera, M. K, Lillibridge.m and Maccormick, “Block-Level Security for Network-attached disks”, In Proc. The 2nd Usenix conference on File and Storage Technologies, pp.159–174, 2003.
10. Anitha, R, Pradeeban Paramjothi, and Saswati Mukherjee, “Security as a Service using Data Steganography in Cloud Computing”, in Proc. of International Conference on Cloud Security Management, pp. 81-89, 2013.
11. Sujitha. G, Rajeswaran, Thiagarajan, Vidya. K, Mercy Shalinie. S, “Preserving Privacy of Cloud Data Using Homomorphic Encryption in MapReduce, “International Journal of Hybrid Information Technology, vol. 7, no. 3, pp. 363-376, 2014.
12. C. Orencik, E. Sava, Efficient and Secure Ranked Multi-Keyword Search on Encrypted Cloud Data, in Proc. OfEDBT- ICDT, pp.186 -195, ACM: New York, USA,2012.

Author Profile

D. Chandravathi is currently working as a Assistant Professor at Gayatri College for Degree and PG courses .Visakhapatnam. Specilaization: Cryptography and Network Security, Bioinformatics.

Dr P.V.Lakshmi is currently working as a Professor at Gitam University,Viskhapatnam.Specilazation: Cryptography and Network Security, Bioinformatics.