ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

Homomorphic Encryption:Security for Cloud Computing

Sanghpriya R.Bangar[1], S.M.Bansode[2]

M. Tech. Student, Department of CSE, SGGSIE&T, Nanded, India[1]

Asst. Professor, Department of CSE, SGGSIE&T, Nanded, India[2]

ABSTRACT: The outsourcing of large amount of data on cloud is increasing day by day. Cloud provides its services pay per demand which is more flexible environment for cloud users. As data is stored onto cloud and perform computations on this data by cloud service provider's (CSP),consequently this poses a security issue. In this paper we discuss the approach which securely stores data in cloud environment. We focus on use of encryption scheme and provide the comparison between them through implementation.

KEYWORDS*:* cloud computing security,homomorphic encryption schemes.

I. INTRODUCTION

Cloud computing is storing of data on remote servers, and accessing them via the internet rather than saving it or installing them on personal or office computer. People don't have idea that they use cloud every day. For example Gmail Google drives which stores data of client. CSP provides services such as *software as a service (SAAS), Platform as a service (PAAS), Infrastructure as a service (IAAS).* The first practical concept of cloud computing came into existence in 2002 by Amazon Web Services. To store data and important information, many users and organizations uses cloud. As user not only wants to secure data in cloud but also while uploading and retrieving data. Major threat of security is as cloud is third party which may monitor user and services used by them which contains the private information [1]. For example in hospital to manage the patient record which contains confidential data, hospital may use cloud. In such case we have to keep data confidential from cloud service providers and to provide better security which is major challenges of cloud service provider's.

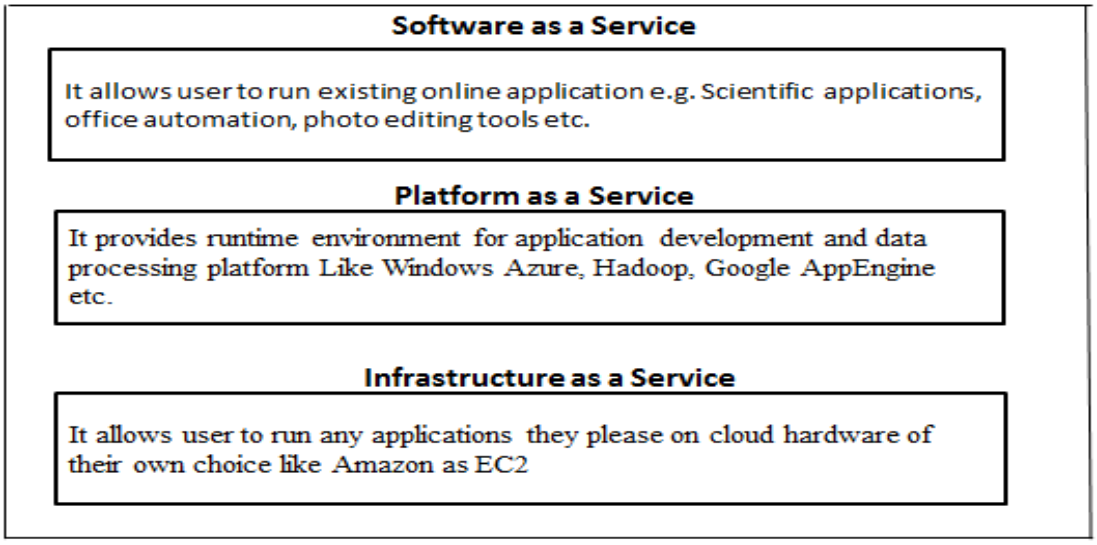

Fig.1.Cloud computing services

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

To provide the solution to the above security challenges, many techniques are proposed and research is still going on. In this paper we proposed one of such technique which provides solution to cloud service provider's challenges.

In section 2, background of homomorphic encryption is described. Shortcomings of conventional encryption and how homomorphic schemes are evolved is explained in section 3. In section 4, definition of homomorphic cryptosystem and its types with algorithms are explained in detail. In section 5, implementation and experiment results against different homomorphic encryption types with time parameter are described. In section 6, At last we provide conclusion of paper.

II. RELATED WORK

There are two types of encryption algorithms:

1) Symmetric key cryptography:Single key is used for encryption and decryption. Ex. Advanced Encryption Standard(AES), Data Encryption Standard(DES) [2], etc.

2) Asymmetric key cryptography:Also known as public key cryptography where different keys(private and public) is used for encryption and decryption. Ex. RSA,homomorphic encryption etc.

In early days symmetric key cryptography techniques are used for uploading, storing and retrieval of data. To provide better security, RSA (Rivest, Adelman and Dertouzos) proposed which is a asymmetric key cryptography in 1978. After that in 1982,Shafi Goldwasser and Silvio Micali proposed an encryption system which provides security by additive homomorphic encryption but it perform operations on single bit .Same concept with security is given by Pascal Paillier was given in 1999 in additive homomorphic encryption. In 2005, Dan Boneh, Eu-Jin Goh and Kobi Nissim proposed security encryption called as somewhat homomorphic encryption which is based on number of additions but only one multiplication is performed. Multiplication is more expensive than addition operation as it takes more execution time [3].

III. PROBLEM STATEMENT

Cloud is the outsourcing of large amount of data so it is lacking of security and confidentiality. It is not sufficient that cloud provides services (SAAS, PASS, IAAS) if it is not providing guaranty of better confidentiality and security to client's data.To secure the cloud means providing security while performing calculations, transferring of message and data storage which is monitor by CSP.

For secure communication private key is exchanged between sender and receiver and message is encrypted by sender's public key and decrypts the cipher text by private key. So private key is responsible to get the original text and if it is lost then there is no use of cipher text. Random Decryption Algorithm (RDA) is proposed in 1978.This technique is able to perform operations on encrypted data stored on cloud. The result is then decrypted to get original text. Here we have to compromise with privacy while performing complex computation on encrypted data.

To solve such issues homomorphic encryption is evolved. Basically homomorphic encryption is used for better security measures. Homomorphic encryption allows performing calculations on data in an encrypted state[4].

IV. HOMOMORPHIC ENCRYPTION

A. Existing approch:

With the existing cryptographic algorithm, we encrypt data before sending to cloud but while performing calculations, for every operation we have to decrypt data[5]. Until homomorphic encryption get evolved, it was impossible to secure encrypted data as to perform different operations on this data client have to send secret key with this data. So data is no more secure and confidential in cloud.

B. Proposed approch:

In this paper we are proposing an application of a method to execute calculations on encrypted data without decrypting them, which will provide the same results after calculations same as worked directly on the original data. Here client is the only holder of private key which ensures privacy of data.

In homomorphic encryption suppose Enc (a) and Enc (b) encryption Enc (f (a, b)), where f can be any operation +, × without the use of private key [5].Here are some functions related to homomorphic encryption algorithm:

1) Key generation function-In this we generate the public key (P_k) and secret key (S_k).

2)Encrypt function -It has public key and message m as input and cipher text (C) as output.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

3)Decrypt function-It takes secret key and cipher text as input and original message (m) as output [11].

C. Categories of Homomorphic Encryption:

According to the operations perform on data; there are categories of homomorphic encryption schemes:

1) Partially homomorphic encryption:Homomorphic encryption which allows performing either addition or Multiplication on encrypted data.

1.1) Additive Homomorphic Encryption Systems:

HomomorphicEncryption is additive, if there is an algorithm that can calculate *Enc(x +y)* from *Enc(x)* and *Enc (y)* without knowing *x* and *y* [6] such as Paillier and Goldwasser-Micali algorithms [7].

Algorithm of Paillier is as follows:

1) Key Generation:

- Choose large prime numbers p and q.
- Let n=p .q such that GCD(n,(p-1)(q-1)=1
- Let Carmichael's function(λ),
  $\lambda = (p-1)(q-1)/GCD((p-1),(q-1))$
- Select generator g in two ways($g \in Z_{n^2*}$)
  Z_n-set of integer n
  Z_{n*}-set of integers co-prime to n. ($\phi(n)$)
  Z_{n^2*}-set of integers co-prime to n^2 such that $n^2 = n(\phi(n))$
  $GCD(g^{\lambda} \bmod n^2 - 1/n, n) = 1$
- $u=(L(g^{\lambda} \bmod n^2))^{-1} \bmod n^2$
  Public Key (n, g) and private key (λ, u).

2) Encryption:

- $m \in Z_n$, $r \in Zn*$

  $C=g^m . r^n \bmod n^2$

3) Decryption:

- $C \in Z_{n2*}$

  $m=(C^{\lambda} \bmod n^2)^{-1} /n .u \bmod n$

Fig.1.Pillier algorithm

The additive homomorphic property of RSA scheme is as follows [12]:

Given $c_1=g^{m1}.r_1^n \bmod n$, $c_2=g^{m2}.r_2^n \bmod n$

$c_1 . c_2=E_{pk}(m_1).E_{pk}(m_2)$

$c_1 . c_2=g^{m1} r_1^n . g^{m2} r_2^n \bmod n$

$=g^{m1+m2}(r1.r_2)^n \bmod n$

$=E_{pk}(m_{1+}m_{2,}r_1.r_2)$ …… [5]

Algorithm of RSA is as follows:

The multiplicative homomorphic property of RSA scheme is as follows [12]:

Given $c_1=m_1^e \bmod n$, $c_2=m_2^e \bmod n$

$c_1 . c_2=E_{lk}(m_1).E_{pk}(m_2)$

$c_1 . c_2=m_1^e . m_2^e \bmod n$

$= (m1.m_2)^e \bmod n$

$=E_{pk}(m_1.m_2)$ …… [13]

1.2) Multiplicative Homomorphic Encryption Systems:

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

A Homomorphic Encryption is multiplicative, if there is an algorithm that can calculate *Enc(x × y)* from *Enc (x)* and *Enc (y)* without knowing *x* and *y* [6] such as RSA and ElGamal Algorithms [8].
Algorithm of RSA is as follows:

1) Key Generation:
- Select prime numbers p , q .
- Compute n = p q, (n) = (p -1). (q -1)
- Select d such that GCD(d,(p-1).(q-1)) = 1
- ϕ (n) = ϕ (p). ϕ (q)=(p-1).(q-1)
- Choose e such that 1<e< ϕ (n)
- e . d=1(mod (ϕ (n)))

Public key (e, n) and Secret key (d)

2) Encryption :
- $C=m^e \bmod n$

3) Decryption:
- $m=C^d \bmod n$

Fig. 3.RSA algorithm

1.3) Additive and Multiplicative Homomorphic Encryption Systems:
These systems allow arbitrary many homomorphic computations of one type and limited number of operations of the other type i.e. it allows both addition and multiplication operations but is not fully homomorphic. An example of this kind would be Boneh-Goh-Nissim cryptosystem [9]. It supports computation of an unlimited number of additions but at most one multiplication.

Algorithm of ElGamal is as follows:

1) Key Generation:
- Select prime number generator b.
- Select random number x.
- Calculates Y:
  $Y= b^x \bmod p$

Public key (b , p and Y) and Secret key (x)

2) Encryption:
- Select a random number r
- Calculates :
  $C=m \cdot Y^k \bmod p$

3) Decryption:
- Select a random number g
- Let d , Calculate
  $d=g^x \bmod p$
- Let e , Calculate
  $e= d^{-1} \bmod p$
- $m=e \cdot C \bmod p$

Fig. 4.ElGamal algorithm

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

2) Fully homomorphic encryption:

All PHE allows to perform either multiplication or addition operation. While Boneh-Goh-Nissim cryptosystem performs only one multiplications and unlimited number of addition.

When we want to perform any type of computation on encrypted data then we go for Fully Homomorphic Encryption (FHE). In 2009 Craig Gentry of IBM developed the first FHE system which enables to perform arbitrary number of multiplications and additions. So that computes all functions on encrypted data in cloud without need of decryption. This achieves the confidentiality of data as cloud doesn't know original text and he is the only holder of secret key [10].The concept of fully homomorphic encryption is impractical.In practical if we implement given homomorphic encryption in Google search, it takes 10 times more time than the simple Google search.

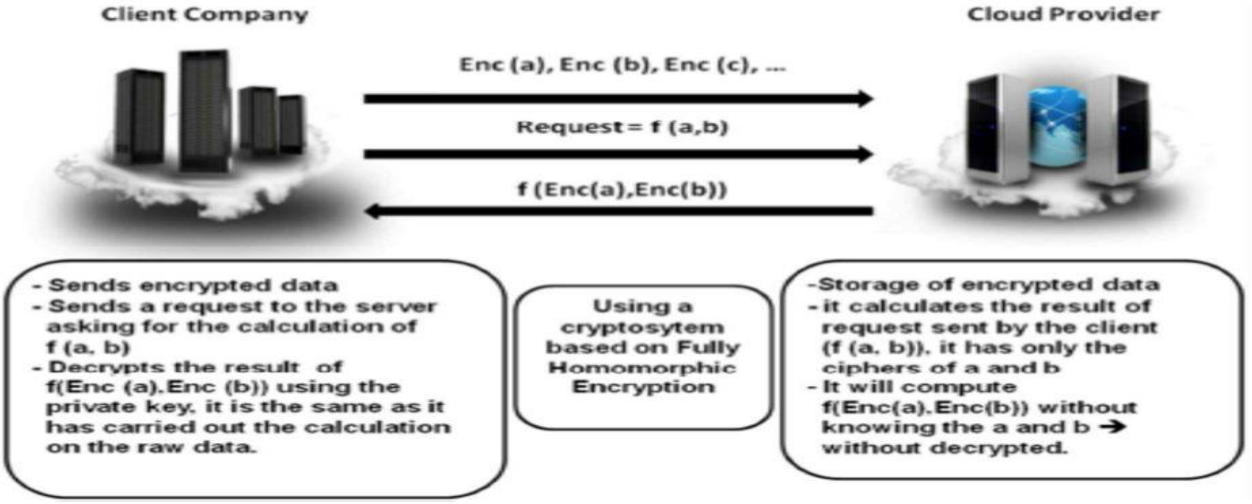

Fig. 5.Fully Homomorphic Encryption applied to the cloud computing Security

V. IMPLEMENTATION AND RESULTS

To provide security in a cloud, different techniques are proposed. In this section we present brief comparison between homomorphic encryption schemes with time complexity parameter which is java based code. Here we take text file or image as input and output is encrypted file along with encryption time. The system specifications used for the experiment are windows 7 on Intel Core i3-350 dual-core 2.26 GHz processor and 4GB DDR3 dual-channel RAM. In these experiments we take text file and image file with different data size against different homomorphic encryption schemes. The code has timer which gives time taken to encrypt and decrypt file. Encrypt time is calculated started when we click on encrypt, till file get uploaded and same with decrypt time calculations. One of the main things we got out from the experiments other than the encryption performance is the size of the encrypted file after getting the cipher text from the plaintext file. This is an important parameter since the larger the file, the more the overhead.

In fig 6. Encrypt and decrypt time of text file is calculated for each homomorphic scheme so that we got idea which encryption type will be better.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

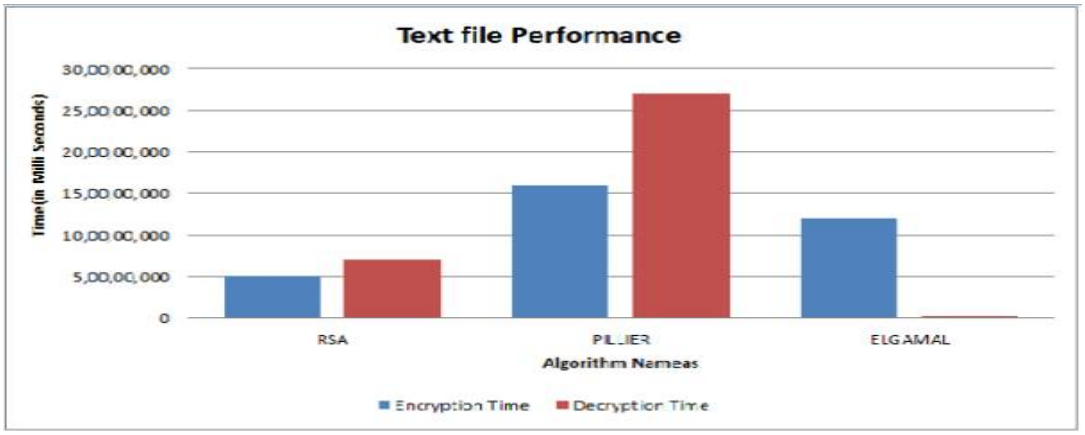

Fig.6. Text file performance

In fig 7. Encrypt and decrypt time of image file is calculated for each homomorphic scheme so that we got idea which encryption type will be better for image type.

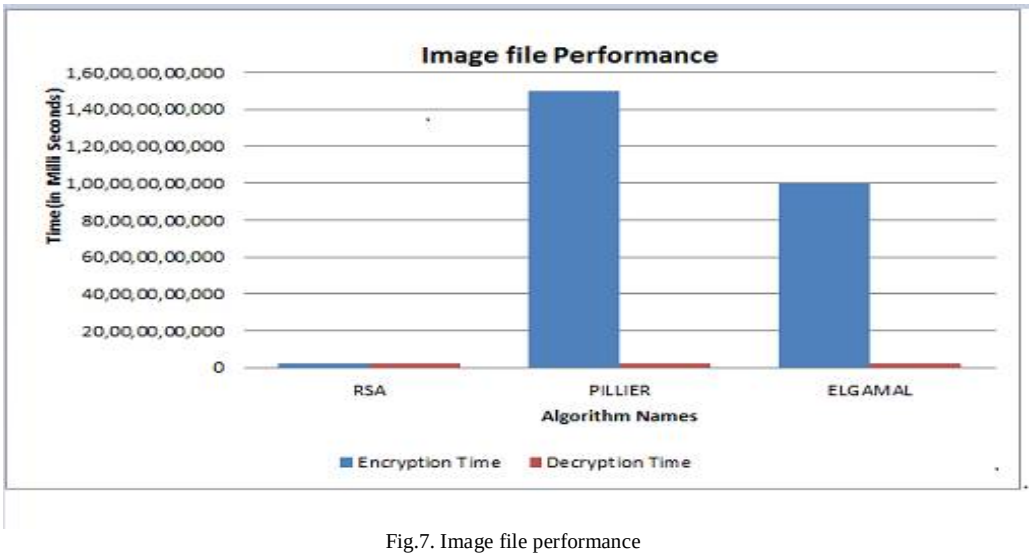

Fig.7. Image file performance

The results are differing for different data file size. Fig.8 shows the homomorphic encryption time over different data file size. Here performance of RSA is exponentially increasing which is better than other algorithms when file size is large.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

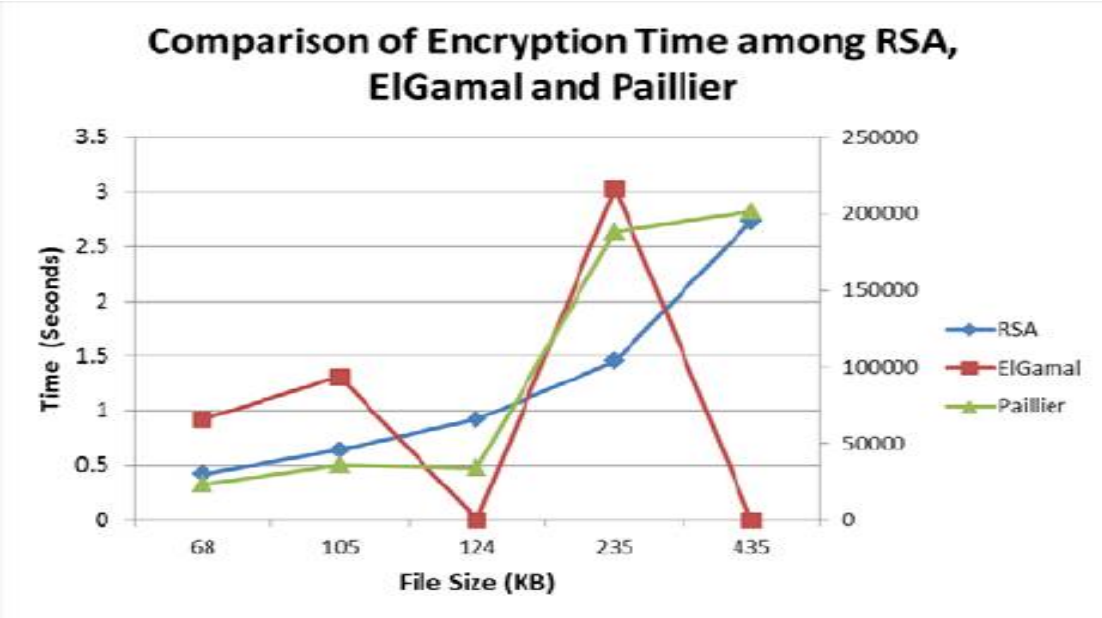

Fig.8. Homomorphic encryption time over different data file size

. Fig.9 shows the homomorphic decryption time over different data file size. Here performance of ElGamal algorithm is better than other algorithms.

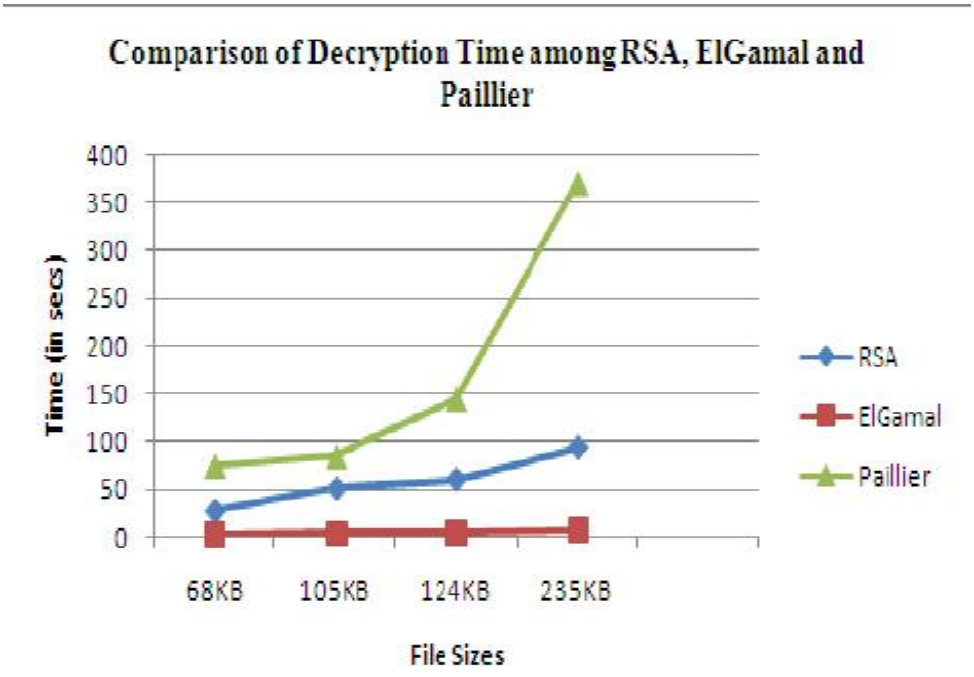

Fig.9. Homomorphic decryption time over different data file size

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

VI. CONCLUSION AND FUTURE WORK

Cloud computing provides low maintenance cost, multi-tenant features. As we handover data to cloud protection issue raises. Here we presented homomorphic encryption technique to provide better security as compared to the conventional encryption scheme. It enable cloud to perform our computations on encrypted data stored on cloud provides result in encrypted form which when decrypted will be same after performing operations on plaintext.

In this paper we described the different homomorphic encryption types with respect to their homomorphic encryption property (only one or mixed properties) and compared their performance on different data size.

In future we will try to optimize performance homomorphic encryption with respect to increasing data size and key maintenance. Work in the area of implementation fully homomorphic encryption is going on.

REFERENCES

1. Hussain Aljafer, Zaki Malik, Mohammed Alodib, Abdelmounaam Rezgui,"A brief overview and and experimental evoution of data confidentiality measures on the cloud",journal of innovation indigital ecosystems I(2014) I-II.
2. Iram Ahmad and ArchanaKhandekar,"International Journal of Information & Computation Technology", ISSN 0974-2239 Volume 4, Number 15 (2014), pp. 1519-1530.
3. Dan Boneh, Eu-Jin Goh, and Kobbi Nissim,"Evaluating 2-DNF formulas on ciphertexts", in Theory of Cryptography Conference, TCC'2005, volume 3378 of Lecture Notes in Computer Science, pages 325-341. Springer, 2005.
4. Fujitsu Labortories Ltd.,"http://www.fujitsu.com/global/about/resources/news/press-releases/2013/0828-01.html".
5. Maha TEBAA, Said EL HAJII, "Secure Cloud Computing through Homomorphic Encryption", International Journal of Advancements in Computing Technology(IJACT) Volume5, Number16, December 2013.R. Nicole, J. Name Stand. Abbrev., in press.
6. Xing Guangli, Chen Xinmeng, Zhu Ping, Ma Jie, "A method of Homomorphic Encryption", Wuhan University Journal of Natural Sciences, Vol.11, No.1, pp.181-184, (2006).
7. Pascal Paillier, "Public-key cryptosystems based on composite degree residuosity classes', 1999.
8. R. Rivest, A. Shamir, and L. Adleman., "A Method for Obtaining Digital Signatures and Public Key Cryptosystems", in Communications of the ACM 21.2, pages 120–126, (1978).
9. D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-dnf formulas on cipher texts",*i*n Proceedings of the Second International Conference on Theory of Cryptography, ser.TCC'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 325–341.
10. Craig Gentry, "A Fully Homomorphic Encryption Scheme", 2009.
11. Yang, Jing, Mingyu Fan, Guangwei Wang, and Zhiyin Kong "Simulation Study Based on Somewhat Homomorphic Encryption.",Journal of Computer and Communications 2 (2014),109.
12.]Eyad Saleh,"Processing Over Encrypted Data: Between Theory and Practice", Proceedings of the 8th Ph. D. Retreat of the HPI Research School on Service-oriented Systems Engineering, (2015).
13.]P. Mell, T. Grance,"The NIST Definition of Cloud Computing", National Institute of Standards and Technology, U. S. Department of Commerce, (2011).

BIOGRAPHY

Sanghpriya R.Bangar is student of Computer network and information security Department, Shri Guru Go bind Singh institute of engineering and technology, Nanded. She received Master of Technology (M.Tech) degree in 2016 from SRTMU, Nanded, MS, India. Her research interests are cloud securityand cache attacks.