International Journal of Advanced Technology in Engineering and Science
Vol. No.5, Issue No. 03, March 2017
www.ijates.com
ijates
ISSN 2348 - 7550

NOISE LEVEL REDUCTION IN FULLY HOMOMORPHIC ENCRYPTION APPLIED TO THE CLOUD COMPUTING SECURITY

Mr.V.Biksham[1], Mr.K.Vijaya Babu[2] Mr.B.Prasad[3]

[1]Associate Professor, [2,3]Assistant Professor, Dept.of CSE, CMR Engineering College, Hyderabad, Telenagana State (India)

ABSTRACT

Homomorphic encryption is a form of encryption which performs arbitrary computations on encrypted data which was when decrypted matches the same results performed on plaintext. In cloud computing we may keep our sensitive data in encrypted format, but if you want do any calculation on cipher text, the key must be share with cloud service providers which may cause to exploit our data. So that to avoid share the key to CSP's instead use the Homomorphic Encryption technique. The computations include searching, sorting, addition, multiplications performed on cipher text. Hence in 2009, Craig Gentry implement a fully homomorphic encryption based on bootstrapping over partially homomophic encryption by using ideal lattices. Now based on literature survey, we conclude that FHE is works theoretically satisfied but practically it may having noise related hurdles and reduce the performance if no of computations may increase. In this paper we propose a new method to reduce noise levels at both partially and Fully Homomophic Encryption techniques. In addition to that distinguishes the noise levels at partially and Fully Homomophic Encryption by various computations.

Keywords: Cloud Computing, Computations, Homomorphic Encryption, Cryptography, Security

I. .INTRODUCTION

The cloud computing is computing paradigm which gives various services and applications in the form of utilities to the IT vendors and Individuals over the Internet. The services include software's, tools, networks, operating systems and storages to all type of customers on demand for a period of time via Internet. At present cloud computing is plays a vital role in development of various products and applications of IT industry with low cost and high quality. It can be come up with public, private and hybrid. The Cloud computing provides various services to the IT and non IT vendors which include service models which are SaaS, IaaS, PaaS, XaaS by using virtualisation [1]. It is the "creation of a virtual (rather than actual) version of something, such as a server, a desktop, a storage device, an operating system or network resources". Creation of a virtual machine over existing operating system and hardware is known as Hardware Virtualization. Virtualization is a technique, which allows sharing a single physical instance of a resource or an application among multiple customers and organizations. It does by assigning a logical name to a physical storage and providing a pointer to that physical resource when demanded. The figure 1 shows the architecture of cloud computing [2] which dynamically

scalable shared resources accessed over a network. It shows the delivered the resources over the Internet with low cost and rent based instead of buying.

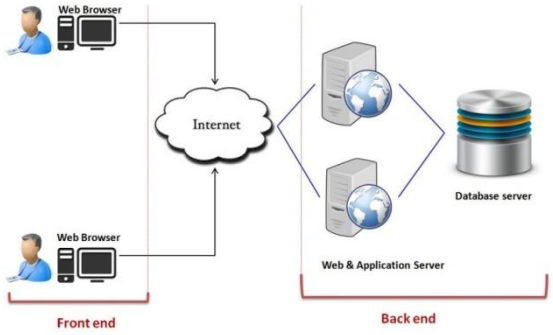


Figure 1: Cloud computing architecture

The following table: I show the top 10 vendors providing cloud services that developed their own cloud service for the benefit of their own and sold it for on demand. Source:

S.No	CSP	Cloud Name	Application
1	IBM	IBM Smart Cloud	SaaS ,PaaS ,IaaS
2	Amazon	EC2	PaaS ,IaaS
3	Google	Google Apps	PaaS, IaaS
4	Microsoft	Azure	PaaS, IaaS
5	Cisco	Webex	NaaS
6	Face book	Face book	PaaS
7	Sales force	Salesforce.com	PaaS
8	Oracle	Oracle	SaaS, DaaS
8	Adobe	Adobe Cloud	SaaS
9	Verizon	Verizon	NaaS
10	Data Direct Tech	Data Direct	DaaS

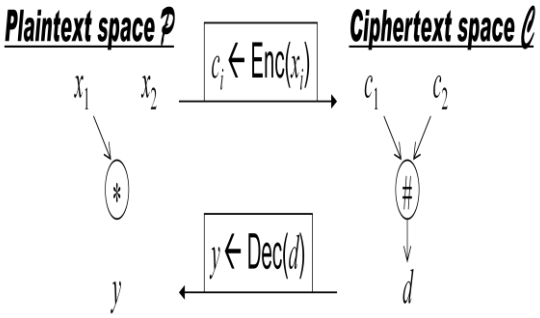
Table1: List of top 10 cloud service providers

The following section II. Related work covers the various existing homomorphic encryption algorithms and their usage and applications with pros and cons are discussed, III covers challenging issues in homomoprhic encryption and problem definition, section IV gives implantation details and results are compared and with other systems which produce the throughput of the algorithm, finally conclusion and future scope observed in section V and followed by various references.

II. RELATED WORK

All the data stored in the cloud were encrypted that would be effectively solve issues like Availability, Data security and third-party control however a user would be unable to leverage the power of the cloud to carry out computation on data without first decrypting it or shipping entirely back to the user for computations[3]. The cloud provider has to decrypt the data first to perform the computation then send the analysis to the user, if the user could carry out any arbitrary computation on the hosted data without the cloud provider learning about the users data computation on encrypted data without prior decryption. Homomorphic encryption [3] schemes allow

the transformation of cipher texts of message m to cipher texts of a computation function of message m without disclosing the message. The security for data and algorithms become very strong due to vast growth of technology variety of attacks on digital goods and technical devices are enabled. Storing and reading data securely there exist several possibilities like secure data encryption. The problem becomes more complex when asking for the possibility to compute with encrypted data or to modify functions are still executable while the privacy is ensured homomorphic cryptosystems can be used.



Some examples:

- "Raw RSA": $c \leftarrow x^e \bmod N$ ($x \leftarrow c^d \bmod N$)
  - $x_1^e \times x_2^e = (x_1 \times x_2)^e \bmod N$
- GM84: $\mathrm{Enc}(0) \in_R$ QR, $\mathrm{Enc}(1) \in_R$ QNR (in Z_N*)
  - $\mathrm{Enc}(x_1) \times \mathrm{Enc}(x_2) = \mathrm{Enc}(x_1 \oplus x_2) \bmod N$

The homomorphic concept was first presented by Rivest Adle-man and Detouzos[RAD78] [4] after the invention of RSA. They asked for an encryption function that permits encrypted data to be operated on without preliminary decryption of the operands. Even in 1978 important matter it is even more important while the partially homomorphic properties of schemes like RSA, Pailler, ElGAmal etc have been acknowledged nor before 2009 when a young IBM researcher Criag Gentry published the first working homomorphic cryptosystem based on lattices [5] and bootstrapping[6] . Homomorphic encryption methods are divided according to the operation to assess on raw data.

A. Partially Vs Fully Homomorphic Encryption

The examples listed below in table1 allow homomorphic computation of some operations on ciphertexts (e.g., additions, multiplications, quadratic functions, etc.).

Cryptosystem	**Partially**	**Fully**
RSA(1977)	Multiplicative	--
ElGamal(1985)	Multiplicative	--
Goldwasser-Micali(1982)	Multiplicative	--
Benaloh(1994)	Multiplicative	--
Paillier(1999)	Addictive	--
Okamoto-Uchiyama(1998)	Integers Modulo	--
Naccache-Stern Knapsack (1998)	Addictive	--
Damgard-Jurik(2001)	Addictive	--
Boneh-Goh-Nissim(2010)	Multiplicative	--

Ishai-Paskin(2006)	Addictive	--
Gentry's cryptosystem (Using Bootstrapping/Ideal Lattices) -2009	--	√
Marten van Dijk, Craig Gentry, Shai Halevi and Vinod Vaikuntanathan (Using integers) -2010	--	√
Brakerski-Gentry-Vaikuntanathan cryptosystem (BGV) (based on the hardness of the Learning with errors problem)	--	√

Table1: List of Homomopric Encryption algorithms

A public key encryption scheme E = (KeyGen, Enc, Dec) is homomorphic for all k and all (pk,sk) output from KeyGen(k) it is possible to define groups M, C so that the plaintext space M, and all ciphertexts output by Encpk are elements of C. for any m1, m2 € M and c1, c2 € C with m1= $Deck_{sk}$ (c1) and m2 = Dec_{sk} (c2) holds that Dec_{sk} (c1 * C2) = m1 * m2. Homomorphic encryption scheme is PKS with the additional property that there exists an efficient algorithm Eval to compute an encryption of the sum or the product of two messages given the public key and encryption of the message. Homomorphic is able to output a ciphertext that encrypts f(m1,,,,mt) where f is any desired function must be efficiently computable information about m1,….mt plain text values should leak.

Data modification and data interruption is an issue in cloud computing public key enables IPSsec of SSL for secure connections. IPSec provide confidentiality and authenticity while SSL protocol generate end to end encryption communication channel between user and host but also from host to host IPsec is compatible with any application and requires IPsec client while SSL is built into every browser. Certifying physical infrastructure servers virtual servers environment users and network devices builds the necessary strong credentials for all virtual entities in the cloud. The integrity and availability of systems need to safeguard against a number of threats include attackers rival competitors terrorists and foreign governments. Growth of wired or wireless communication networks force the clouds to be secured through firewalls intrusion detection and prevention system encryption. Intrusion detection and prevention systems can early detect the malicious activity and prevent the serious to the cloud for prevention & detection can be used as forensic evidence which can used in legal terms.

III. PROBLEM DEFINITION

Cloud computing is a technology that uses on internet, central remote servers to maintain data and applications that allows consumers and business to use applications without installation. Cloud can be integrated with other systems like SAP, SOA, People soft transfers the files from one system to other system through the middleware technology with server access and adapter type such as soap file proxy. Cloud computing classified into three segments application storage and connectivity each one is used for different purpose and offers different products for business and individuals around the world, which uses remote server to maintain data and various

applications provides significant cost effective IT resources as cost on demand IT based on the actual usage of the customer. Cloud computing provides clients with a virtual computing infrastructure on top of which they can store data and run applications, to transfer data from one device to other device we design cryptographic primitives and protocols to the setting of cloud computing attempting to strike a balance between security efficiency and functionality. Cloud computing infrastructure do not provide any security against entrusted cloud operators making them unsuitable for storing sensitive information such as medical records financial records for high impact business data[7].

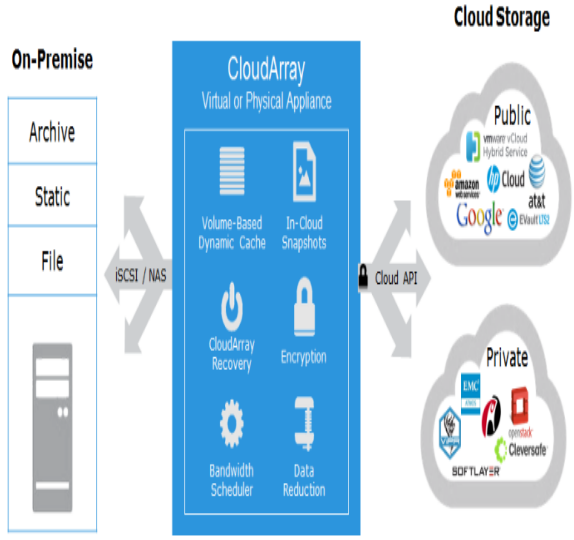


Figure 2 Data transfer from Cloud Array to cloud API

Cloud array can join the conversation by providing a bridge point between traditional storage arrays and less expensive cloud storage. Data identified for archive or cold can be moved across to cloud array then acts as a local cache for object based storage resources provided by public or private cloud. Cloud array integrates a number of features that really punch above what one would except from a little blue box including data compression dynamic resizable caches snapshot and secure encryption of data. The most common use of encryption is to provide confidentiality by hiding all useful information about the plaintext. Encryption renders data useless in the sense that one loses the ability to operate on it; we design cryptosystems that support a variety of computations on encrypted data ranging from general purpose computations to special purpose computations.

A. Homomorphic Encryption to Cloud computing Security

The data transferred to the cloud use standard encryption methods to secure the operations and the storage of the data, to encrypt the data before send it to the cloud provider. The client will need to provide the private key to the server to decrypt data before execute the calculations required which might affect the confidentiality and privacy of data stored in the cloud. Homomorphic Encryption systems are used to perform operations on encrypted data without knowing the private key, the client is the only holder of the secret key when we decrypt the result of operation is the same if we carried out the calculation on the raw data.

a). BGV Encryption scheme:

The integer vectors whose security is dependent on the hardness of decisional Learning LWE[8] and dealing with the integer polynomials[9] is dependent on the hardness of the decisional Ring LWE are two versions of the cryptosystem. BGV is an asymmetric encryption technique which can be used for the encryption of the bits.

Encrypt (Plaintext m, PublicKey Priv);

Decrypt (ciphertext c, PrivateKey Priv);

Level shifting operations

Rescale (Ciphertext c); Ciphertext c

Switch Key (Augmented Ciphertext c); Ciphertext c

Homomorphic operations

Add (Ciphertext c1, Ciphertext c2); Ciphertext csum

Mul (Ciphertext c1, Ciphertext c2); Ciphertext cmul

b) Gortis Enhanced Homomorphic Cryptosystem:

Enhanced cryptosystem used for homomorphic encryption or decryption with IND-CCA secure [10]. The numerous applications of the type of homomorphic encryption has the basic concept that the computer will perform the computations on the already encrypted data without any knowledge of the real value and the encrypted message or data will be sent back as a result and decrypted.

The decrypted message must be equal to the intended computed value if performed on the real data for the particular structure has to be presented by the encryption.

Choose large prime number p and another prime number q

Calculate m=p*q

Generate a random number q

Key: p

Encryption (X, m, p, q, r)

Assume X€Z_p

Compute Y=(X+r*pq) (mod m)

Output Y€Z_c

Decryption

Decrypt (Y, P)

Input Y€Zc

Compute X=Y mod P output XZp

c. Non-interactive Exponential Homomorphic Encryption Scheme:

Non-interactive encryption exponential functions and polynomial functions can be implemented in the following way. Describe the problem of non-interactive evaluation of encrypted functions.

Alice The originating host has an algorithm to compute a function f

The remote host has an input x and is willing to compute f(x)

Alice wants bob to learn nothing substantial about f, the bob should not need to interact with Alice during the computation of f(x).

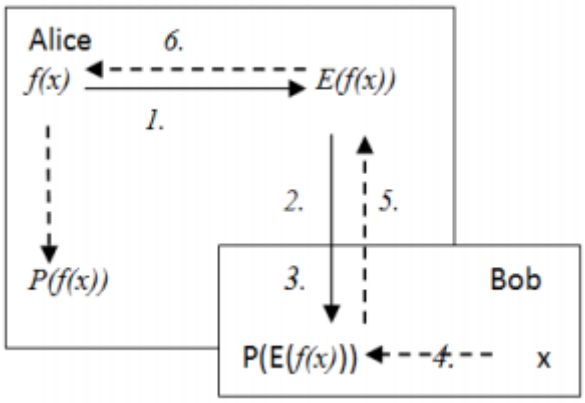


Figure 3 Gorits Enhanced Homomorphic Cryptosystem

1. Alice will encrypt f.
2. Alice will creates a program P(E(f))) which implements E(f).
3. Alice will send P(E(f)) to bob.
4. Bob will send P(E{f))(x) at x.
5. Bob will send P(E(f))(x) to f(x).
6. Alice will decrypt P(E(f))(x) and obtain f(x).

A public key algorithm is reviewed in order to prove the exponential homomorphic encryption is security based on the difficulty of factoring large integer.

d. Algebra Homomorphic encryption Scheme based on Updated EIGamal:

The modified form of the digital signature standard DSS presented by the NIST in America the security of the AHEE is IND-CPA which is the highest level of the security of AHEE. Homomorphic of this algorithm refers the same k for encryption but uses the random number of k in E190 which makes AHEE able to resist plaintext attack proved to be secure. Sander and Tshudin defined additive and multiplicative homomorphic on integer ring operations.

Select any two prime numbers say p and q, Calculate the product of those two prime numbers say N = p*q where p and q being confidential and N is public.

Select random number x and a root g of GF (p).where g and x are smaller than p.

Calculate y = g x mod p. use this y for the encryption

Encryption will be performed in the following ways

Select random integer number r and apply following homomorphic encryption.

E1 (M) = (M+r*p) mod N.

Select random integer number k and the encryption algorithm are

E_g (M) = (a,b) = (g^kmod p, y^k E_1 (M) mod p)

Decrypted algorithm $D_{g\ () \text{ is } M =}$ b x (a^{x-1}) (mod p).

e. Data security and privacy using Homomorphic Encryption in cloud:

Authorization is used to control the access of data mechanism by which a system determines level of access a particular authenticated user should have to secure resources controlled by the system. Data integrity in the cloud system preserving information integrity should modified by unauthorized users, it provide cloud computing service such as SaaS PaaS and IaaS of large-scaled data, cloud computing environment usually

provides data processing. Verifying the integrity of data in the cloud remotely is the perquisite to deploy applications proof of retrievability to realize the remote data integrity checking by combining error correction code and spot checking.

Data confidentiality users to store private and confidential data in the cloud authentication and access control strategies are used to ensure data confidentiality. Encryption is used to ensure the confidentiality of data homomorphic encryption is system proposed by Rivest et al ensures that the cipher text algebraic operation results are consistent with the clear operation after encryption system involves complicated calculation and the cost of computing and storage is very high. Data concealment used to put the data confidentiality in the cloud Delettre et al [11] proposed a concealment concept for databases security which combines real data with the visual abnormal data to false rate the real data volume. The authorization users can easily separate the abnormal data from the real data. Data concealment techniques increase the overall volume of real data provides enhanced security for the private data, the technique concealment is to make the real data safe and secure from malicious users and hackers. Data privacy is the ability of an independent group to select themselves a subject may be more concerned about the current or future information being revealed then information from the past. A user may be comfortable friends can manually request for their information but the user may not like alerts to be sent automatically, rather their information reported as an ambiguous region rather than a precise point.

IV. IMPLEMENTATION

A. Noise levels of Homomorphic Encryption in Cloud storage

The noise level in homomorphic encryption applied to the cloud as in addictive and multiplicative. Here we analyze the noise levels after applying multiple computations on encrypted data. The noise[12] level depends on the number of computations carried on encrypted data.

i) XORing two encrypted bits:

Secret key: large odd number p

To Encrypt a bit b:

pick a (random) “large” multiple of p, say $q \cdot p$

pick a (random) “small” number $2 \cdot r+b$

(this is even if b=0, and odd if b=1)

Ciphertext $c = q \cdot p+2 \cdot r+b$

To Decrypt a ciphertext c:

Taking c mod p recovers the noise

$c_1 = q_1 \cdot p + (2 \cdot r_1 + b_1)$

$c_2 = q_2 \cdot p + (2 \cdot r_2 + b_2)$

$c_1+c_2 = p \cdot (q_1 + q_2) + 2 \cdot (r_1+r_2) + (b_1+b_2)$

Odd if $b_1=0$, $b_2=1$ (or)

$b_1=1$, $b_2=0$

Even if $b_1=0$, $b_2=0$ (or)

$b_1=1$, $b_2=1$

$c_1+c_2 = p \cdot (q_1 + q_2) + 2 \cdot (r_1+r_2) + (b_1+b_2)$

lsb= b_1 XOR b_2

the noise grows!

$c_1+c_2 = p\cdot(q_1 + q_2) + 2\cdot(r_1+r_2) + (b_1+b_2)$

noise= 2 * (initial noise)

Noise is 17

ii) Multiplying two encrypted bits:

$c_1c_2 = p\cdot(c_2\cdot q_1+c_1\cdot q_2-q_1\cdot q_2) + 2\cdot(r_1r_2+r_1b_2+r_2b_1) + b_1b_2$

noise = (initial noise)^2

NOISE grows 17+14=31 (increase 14) (decryption noise recover is 3)… so what's the problem? If the |noise| > p/2, then … decryption will output an incorrect bit

V. RESULTS & DISCUSSIONS

The noise level at addictive or Somewhat Homomorphic Encryption[13] is equals to the two times of initial noise where as in Fully Homomorphic Encryption or multiplicative is square of the initial noise level. The following graph shows the noise levels at both SHE and FHE with x-axis shows the size of ciphertext and y-axis shows the increase of number of computations on ciphertext as shown in the figure 4.

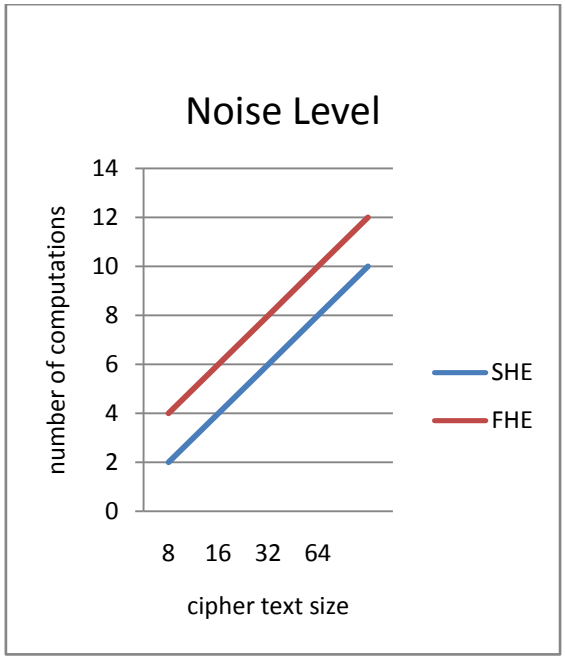


Figure 4: Noise levels at SHE and FHE

VI. CONCLUSION

Homomorphic cryptosystems allow for the same level of privacy as any other cryptosystem, while also allowing for operations to be performed on the data without the need to see the actual data. If a computationally efficient fully homomorphic cryptosystem were to be developed, the implications are phenomenal. Complete privacy

between client and server would be possible without any decreased functionality. Such systems could be applied to nearly anything that requires computation, such as voting, banking, cloud computing, and many others.

REFERENCES

[1.] Secure virtualization for cloud computing,*Journal of Network and Computer Applications*, Volume 34, Issue 4, Pages 1113-1122,July 2011, http://dx.doi.org/10.1016/j.jnca.2010.06.008.

[2.] http://www.cepis.org/index.jsp?p=641&n=825&a=4758

[3.] Aderemi A. Atayero*, Oluwaseyi Feyisetan**,*Covenant University, Nigeria, Email:atayero@ieee.org** Kings College, UK, Email:oluwaseyi.feyisetan@kcl.ac.uk

[4.] "Security Issues in Cloud Computing:The Potentials of Homomorphic Encryption" V. 2, NO. 10, Oct-2011 SSN 2079-8407

[5.] Brakerski, Zvika, and Vinod Vaikuntanathan. "Lattice-Based FHE as Secure as PKE." Proceedings of the 5th Conference on Innovations in Theoretical Computer Science - ITCS '14 January 12–14, 2014, Princeton, New Jersey, USA. ACM, p.1-12.

[6.] Craig Gentry IBM,Shai Halevi IBM,Nigel P. Smart University of Bristol "Better Bootstrapping in Fully Homomorphic Encryption" December 15, 2011, https://eprint.iacr.org/2011/680.pdf

[7.] Paul Hofmann ,VP, Group of the Chief Scientist, SAP Labs Palo Alto, SAP Labs, LLC "The Limits of Public Clouds for Business Applications" An overly simplistic reliance on the utility model risks blinding us to the real opportunities and challenges of cloud computing © Atlantis Press, 2011,https://www.researchgate.net/profile/Paul_Hofmann6/publication/220490938_Cloud_Computing_The_Limits_of_Public_Clouds_for_Business_Applications/links/544495330cf2e6f0c0fbaa0f.pdf

[8.] Gentry, Craig, Shai Halevi, and Vinod Vaikuntanathan. "A simple BGN-type cryptosystem from LWE." *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer Berlin Heidelberg, 2010.

[9.] Liam Morris, "Analysis of Partially and Fully Homomorphic Encryption" l m1115@rit.edu,Department of Computer Science, Rochester Institute of Technology, Rochester, New York May 10, 2013.

[10.] Ronald L. Rivest Len Adleman Michael L.Dertouzos "ON DATA BANKS AND PRIVACY HOMOMORPHISMS"*lucagiuzzi.unibs.it/corsi/Support/papers,* Copyright © 1978 by Academic Press, Inc. *cryptography/RAD78.pdf*

[11.] Oded Regev, "On lattices, learning with errors, random linear codes, and cryptography," in Proceedings of the thirty-seventh annual ACM symposium on Theory of computing (Baltimore, MD, USA: ACM, 2005), 84-93, http://portal.acm.org/citation.cfm?id=1060590.1060603.

[12.] G.Asha Prabha1 et.all "Ensuring Distributed Accountability for Data Sharing in the Cloud" GKM College of Engineering Technology , Perungalathur, Chennai, India.International Journal of Engineering Science and Computing, March 2016 2508http://ijesc.org/DOI 10.4010/2016.592ISSN 2321 3361 © 2016 IJESC.

[13.] Beyond Provable Security Verifiable IND -CCA Security of OAEP Gilles Barthe 1 , Benjamin Gr´egoire 2 , Yassine Lakhnech 3 , and Santiago Zanella Beguelin 1 1 IMDEA Software 2 INRIA Sophia Antipolis-M´editerran´ee 3 Universit´e Grenoble 1, CNRS, Verimag Rochester Institute of Technology, Rochester, New York, http://www.liammorris.com/crypto2/Homomorphic%20Encryption%20Paper.pdf May 10, 2013,

[1]Mr. V.Biksham working as Assoc.Prof in CMR Engineering College, currently he is pursuing Ph.D from JNTUH and having 11 years of experience in teaching, He had published 3 papers in various International Conferences and 8 papers in various International Journals. His areas of interest are Cloud computing and Linux Programming.

[2]Mr. K.Vijaya Babu working as Asst.Prof in CMR Engineering College, He is having 5 years of experience in teaching, He had published 4 papers in various International Journals. His areas of interest are Compute forensics and Computer Networks.

[3]Mr.B.Prasad working as Asst.Prof in CMR Engineering College, He is having 5 years of teaching experience, He had published 3papers in various International Journals. His areas of interest are Distributed Systems and Cloud Computing.