OPEN ACCESS
ISSN: 2249-6645

International
Journal
Of Modern Engineering Research (IJMER)

Computing on encrypted data through homomorphic encryption in cloud security

Mr.V.Biksham[1], Dr.D.Vasumathi[2]

[1]*Mr.V.Biksham, Research Scholar, Dept. of CSE, CMR Engineering College, Hyderabad India.*
E-Mail:vbm2k2@gmail.com.
[2] *Dr.D.Vasumathi, Professor, Dept. of CSE, JNTUniversity, Hyderabad, India.*
E-Mail: rochan44@gmail.

Abstract: Cloud computing is an emerging area where we can access the various resources from service provides via internet with low cost and high efficiency. It solves many problems related to store the huge amount of data, accessing the resources such as software, applications, and platforms to various clients who are developing client server communication through any private or public networks. Today cloud security is become more challenging task to many researcher to give privacy and confidentiality on original data that to be shared among them. To give security to cloud data, the cloud service provide maintain the encryption techniques to secure the data. Apart from that the client can perform any computations on raw data frequently, to do this to every time decrypt the raw data and do the computations on the plain text and update in the cloud. But to avoid the every time decrypting the raw data for computation, we proposes a system called Homomorphic Encryption that performs computation on raw data without decrypting.

Keywords: Cloud Computing, Confidentiality, raw data, Homomorphic Encryption.

I. INTRODUCTION:

Cloud Computing provides us means of accessing the applications as utilities over the Internet. It allows us to create, configure, and customize the applications online. The term Cloud refers to a Network or Internet. In other words, we can say that Cloud is something, which is present at remote location. Cloud can provide services over public and private networks, i.e., WAN, LAN or VPN. Applications such as e-mail, web conferencing, customer relationship management (CRM) execute on cloud. Cloud Computing refers to manipulating, configuring, and accessing the hardware and software resources remotely[1]. It offers online data storage, infrastructure, and application. Cloud computing offers platform independency, as the software is not required to be installed locally on the PC. Hence, the Cloud Computing is making our business applications mobile and collaborative. There are certain services and models working behind the scene making the cloud computing feasible and accessible to end users. Following are the working models for cloud computing: Deployment Models: Deployment models define the type of access to the cloud, i.e., how the cloud is located? Cloud can have any of the four types of access: Public, Private, Hybrid, and Community. The public cloud allows systems and services to be easily accessible to the general public. Public cloud may be less secure because of its openness. The private cloud allows systems and services to be accessible within an organization. It is more secured because of its private nature. The community cloud allows systems and services to be accessible by a group of organizations. The hybrid cloud is a mixture of public and private cloud, in which the critical activities are performed using private cloud while the non-critical activities are performed using public cloud. Service Models**:** Cloud computing is based on service models. These are categorized into three basic service models which are -

Anything-as-a-Service (XaaS) is yet another service model, which includes Network-as-a-Service, Business-as-a-Service, Identity-as-a-Service, Database-as-a-Service or Strategy-as-a-Service. The **Infrastructure-as-a-Service (IaaS)** is the most basic level of service. Each of the service models inherits the security and management mechanism from the underlying model, as shown in the following diagram:

II. CLOUD COMPUTING:

Cloud computing enables companies to consume compute resources as a utility -- just like electricity -- rather than having to build and maintain computing infrastructures in-house. Cloud computing promises several

attractive benefits for businesses and end users. Three of the main benefits of cloud computing includes: Self-service provisioning: End users can spin up computing resources for almost any type of workload demand. Elasticity: Companies can scale up as computing needs increase and then scale down again as demands decrease.

Pay per use: Computing resources are measured at a granular level; allowing users to pay only for the resources and workloads they use. Cloud computing services can be private, public or hybrid. Private cloud services are delivered from a business' data center to internal users. This model offers versatility and convenience, while preserving management, control and security. Internal customers may or may not be billed for services through IT chargeback. In the public cloud model, a third-party provider delivers the cloud service over the Internet. Public cloud services are sold on-demand, typically by the minute or the hour. Customers only pay for the CPU cycles, storage or bandwidth they consume. Leading public cloud providers include Amazon Web Services (AWS), Microsoft Azure, IBM/Soft Layer and Google Compute Engine.

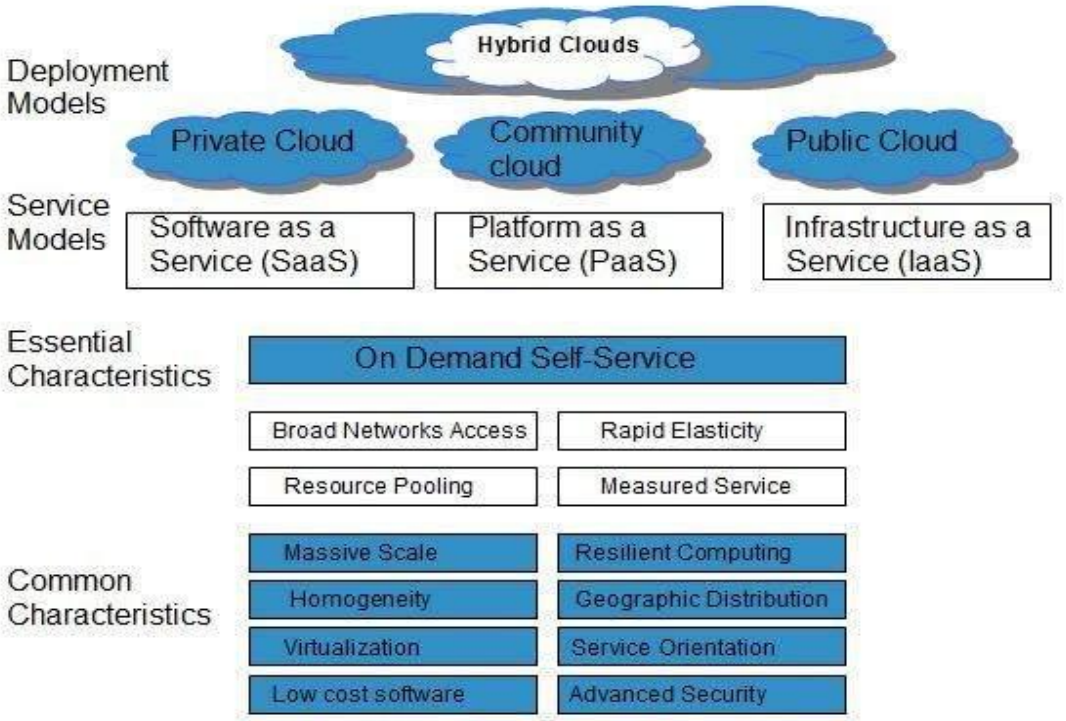

Fig1: Cloud Computing Architecture

Hybrid cloud is a combination of public cloud services and on-premises private cloud – with orchestration and automation between the two[2]. Companies can run mission-critical workloads or sensitive applications on the private cloud while using the public cloud for bursty workloads that must scale on-demand. The goal of hybrid cloud is to create a unified, automated, scalable environment which takes advantage of all that a public cloud infrastructure can provide, while still maintaining control over mission-critical data. Although cloud computing has changed over time, it has always been divided into three broad service categories: infrastructure as a service (IaaS), platform as a service (PaaS) and software as service (SaaS). IaaS providers such as AWS supply a virtual server instance and storage, as well as application program interfaces (APIs) that let users migrate workloads to a virtual machine (VM). Users have an allocated storage capacity and start, stop, access and configure the VM and storage as desired. IaaS providers offer small, medium, large, extra-large, and memory- or compute-optimized instances, in addition to customized instances, for various workload needs. In the PaaS model, providers host development tools on their infrastructures. Users access those tools over the Internet using APIs, Web portals or gateway software. PaaS is used for general software development and many PaaS providers will host the software after it's developed. Common PaaS providers include Salesforce.com's Force.com, Amazon Elastic Beanstalk and Google App Engine. SaaS is a distribution model that delivers software applications over the Internet; these are often called Web services. Microsoft Office 365 is a SaaS offering for productivity software and email services. Users can access SaaS applications and services from any location using a computer or mobile device that has Internet access.

Cloud computing is defined as a type of computing that relies on *sharing computing resources* rather than having local servers or personal devices to handle applications. Cloud computing is comparable to grid

computing, a type of computing where unused processing cycles of all computers in a network are harnesses to solve problems too intensive for any stand-alone machine. In cloud computing, the word cloud (also phrased as "the cloud") is used as a metaphor for "*the Internet*," so the phrase *cloud computing* means "a type of Internet-based computing," where different services — such as servers, storage and applications — are delivered to an organization's computers and devices through the Internet.

II.i How Cloud Computing Works

The goal of cloud computing is to apply traditional supercomputing, or high-performance computing power, normally used by military and research facilities, to perform tens of trillions of computations per second, in consumer-oriented applications such as financial portfolios, to deliver personalized information, to provide data storage or to power large, immersive online computer games. To do this, cloud computing uses networks of large groups of servers typically running low-cost consumer PC technology with specialized connections to spread data-processing chores across them. This shared IT infrastructure contains large pools of systems that are linked together. Often, virtualization techniques are used to maximize the power of cloud computing.

Cloud computing, also known as on-demand computing, is a kind of Internet-based computing, where shared resources, data and information are provided to computers and other devices on-demand. It is a model for enabling ubiquitous, on-demand access to a shared pool of configurable computing resources. Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in third-party data centers. It relies on sharing of resources to achieve coherence and economies of scale, similar to a utility (like the electricity grid) over a network. At the foundation of cloud computing is the broader concept of converged infrastructure and shared services. Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort.

Cloud computing, or in simpler shorthand just "the cloud", also focuses on maximizing the effectiveness of the shared resources. Cloud resources are usually not only shared by multiple users but are also dynamically reallocated per demand. This can work for allocating resources to users. For example, a cloud computer facility that serves European users during European business hours with a specific application (e.g., email) may reallocate the same resources to serve North American users during North America's business hours with a different application (e.g., a web server). This approach helps maximize the use of computing power while reducing the overall cost of resources by using less power, air conditioning, rack space, etc. to maintain the system. With cloud computing, multiple users can access a single server to retrieve and update their data without purchasing licenses for different applications.

The term "moving to cloud" also refers to an organization moving away from a traditional CAPEX model (buy the dedicated hardware and depreciate it over a period of time) to the OPEX model (use a shared cloud infrastructure and pay as one uses it). Proponents claim that cloud computing allows companies to avoid upfront infrastructure costs, and focus on projects that differentiate their businesses instead of on infrastructure. Proponents also claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and enables IT to more rapidly adjust resources to meet fluctuating and unpredictable business demand. Cloud providers typically use a "pay as you go" model. This can lead to unexpectedly high charges if administrators do not adapt to the cloud pricing model.

The present availability of high-capacity networks, low-cost computers and storage devices as well as the widespread adoption of hardware virtualization, service-oriented architecture, and autonomic and utility computing have led to a growth in cloud computing. Companies can scale up as computing needs increase and then scale down again as demands decrease. Cloud computing has now become a highly demanded service or utility due to the advantages of high computing power, cheap cost of services, high performance, scalability, accessibility as well as availability. Cloud vendors are experiencing growth rates of 50% per annum. But due to being in a stage of infancy, it still has some pitfalls which need to be given proper attention to make cloud computing services more reliable and user friendly.

III. DATA PROTECTION IN CLOUD COMPUTING:

Cloud computing is play a vital role in present IT industry. It provides various resources like storage in servers, accessing any plot forms and networks over any network with low cost. The business person who wants to outsource the any services from cloud, it becomes very easy and efficient through a simple connecting with cloud providers' servers. At the same time multiple users can outsource the various services from the Cloud servers without any delay and efficiently concurrent access to the cloud servers. Apart from that security is become more challenging task to many company as well as CSPs to get confidence over the cloud services from various cryptanalysis. Encryption is one of major concept to protect the data of clients over the cloud. The following diagram shows a simple mechanism to provide security from active/passive attacks. The cloud service

provider provides a mechanism called encryption to provide security and privacy to the clients who are outsourcing the cloud services via Internet. The encryption is a process of converting plain text in coded format which is called as cipher text.

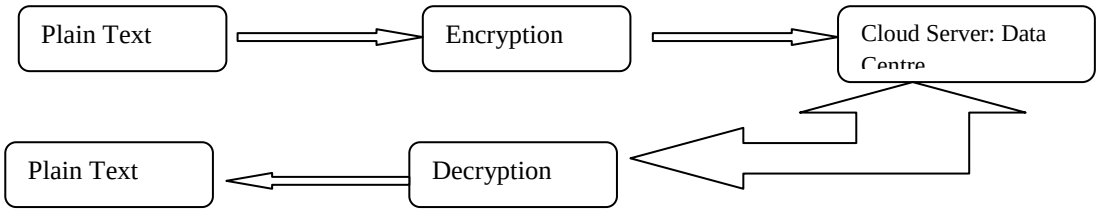

Fig2: Cloud Security

IV. HOMOMORPHIC ENCRYPTION:

Background: The word homomorphic has roots in Greek and loosely translates as same shape or same form. In relation to cryptography, the concept is that operations can be performed on encrypted data without sharing the secret key needed to decrypt the data. Homomorphic encryption has great utility in cloud computing particularly for those that wish to house encrypted data on cloud providers' servers.

Enc (a), Enc (b)…
Request=f (a, b)
F (Enc (a), Enc (b)

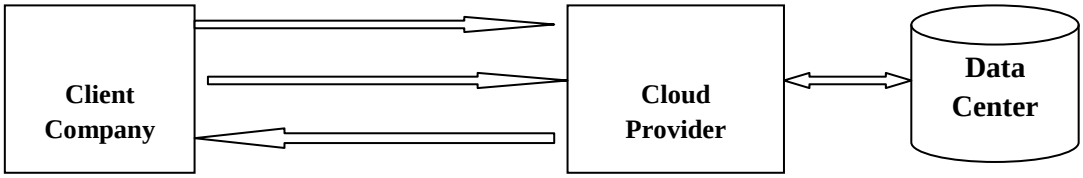

Fig3: Simple Homomorphic Encryption scheme

The main aim of Homomorphic encryption is to enhance the security of cloud computing. The data should be encrypted and sent to the cloud. After it is send, the computations are made on the encrypted data and the result of this computation is an encrypted data too. If the result of the computation is decrypted, then the plain text version of the result is got back. The question is, why it is assumed that it enhances the cloud computing? The data is just encrypted and sent to cloud. When the computations are to be made, these data are queried to the computers and decrypted, made computations, then send back to the cloud if needed. Homomorphic encryption in and of itself solves the problem of computation on encrypted data[3]. For example, Enc (a) and Enc (b) it is possible to compute encryption Enc (f (a, b)) where f can be: +, ×, ÷, Ex-OR and without using the private key as shown in fig. Amount the Homomorphic Encryption we distinguish, according to the operations that allows to assess on raw data, the additive Homomorphic encryption(only additions of the raw data) is the (e.g., Paillier and Goldwasser and Micali) cryptosystems, and the multiplicative Homomorphic Encryption (only products on raw data) is the (e.g. RSA, Elgamal) cryptosystem.

The following table shows various homomorphic cryptosystems which support for addictive and multiplicative functionalities.

Cryptosystem	**Homomorphic Operations**	**Homomorphic Property**
RSA	Multiplication mod n	Multiplicative
Elgamal	Multiplication, Exponentiation	Multiplicative
Paillier	Addition, Subtraction,	Addictive
Goldwasser-Micali	XOR	Addictive
Benaloh	Addition, Subtraction	Addictive
Naccache-Stern	Addition, Subtraction,	Addictive
Gentry FHS	Addition, Subtraction, Multiplication, Exponentiation	Both

Table1: List of Homomorphic Encryption methods

A. Addictive Homomorphic Encryption(Paillier Cryptosystem)

Key generation

1. Choose two large prime numbers *p* and *q* randomly and independently of each other such that this property is assured if both primes are of equal length.
2. Compute n=pq and $n = pq$ and $\lambda = \operatorname{lcm}(p-1, q-1)$.
3. Select random integer g where $g \in \mathbb{Z}^*_{n^2}$
4. Ensure n divides the order of g by checking the existence of the following modular multiplicative inverse: $\mu = (L(g^\lambda \bmod n^2))^{-1} \bmod n$,

Where function L is defined as $L(u) = \frac{u-1}{n}$.

Note that the notation $\frac{a}{b}$ does not denote the modular multiplication of a times the modular multiplicative inverse of b but rather the quotient of a divided by b, i.e., the largest integer value $v \geq 0$ to satisfy the relation $a \geq vb$.

- The public (encryption) key is (n, g).
- The private (decryption) key is (λ, μ).

If using p,q of equivalent length, a simpler variant of the above key generation steps would be to set $g = n+1, \lambda = \varphi(n)$, and $\mu = \varphi(n)^{-1} \bmod n$, where $\varphi(n) = (p-1)(q-1)$

Encryption

1. Let m be a message to be encrypted where $m \in \mathbb{Z}_n$
2. Select random r where $r \in \mathbb{Z}^*_n$
3. Compute ciphertext as: $c = g^m \cdot r^n \bmod n^2$

Decryption

1. Let c be the ciphertext to decrypt, where $c \in \mathbb{Z}^*_{n^2}$
2. Compute the plaintext message as: $m = L(c^\lambda \bmod n^2) \cdot \mu \bmod n$

As the original paper points out, decryption is "essentially one exponentiation modulo n^2."

Homomorphic properties

A notable feature of the Paillier cryptosystem is its homomorphic properties. As the encryption function is additively homomorphic, the following identities can be described:

Homomorphic addition of plaintexts

The product of two ciphertexts will decrypt to the sum of their corresponding plaintexts,

$$D(E(m_1, r_1) \cdot E(m_2, r_2) \bmod n^2) = m_1 + m_2 \bmod n.$$

The product of a ciphertext with a plaintext raising g will decrypt to the sum of the corresponding plaintexts,

$$D(E(m_1, r_1) \cdot g^{m_2} \bmod n^2) = m_1 + m_2 \bmod n.$$

Homomorphic multiplication of plaintexts

An encrypted plaintext raised to the power of another plaintext will decrypt to the product of the two plaintexts,

$$D(E(m_1, r_1)^{m_2} \bmod n^2) = m_1 m_2 \bmod n,$$
$$D(E(m_2, r_2)^{m_1} \bmod n^2) = m_1 m_2 \bmod n.$$

More generally, an encrypted plaintext raised to a constant *k* will decrypt to the product of the plaintext and the constant,

$$D(E(m_1, r_1)^k \bmod n^2) = k m_1 \bmod n.$$

However, given the Paillier encryptions of two messages there is no known way to compute an encryption of the product of these messages without knowing the private key.

B. Multiplicative Homomorphic Encryption(RSA Cryptosystem)

RSA encryption scheme introduced by Rivest,Shamir and Adleman[5] (RSA 78) ha multiplicative homomorphism. Recall that RSA cryptosystem works like this:

- To generate a public key/secret key pair Gen chooses two large prims p and q and sets N=p.q.Gen also choses an integer e co prime to Ø(N)=(p-1)(q-1). The public key pk is (N,e) and the secret key sk is (p,q) we note that given p and q it is easy to calculate $d=e^{-1} \bmod Ø(N$.
- The set of messages consist of all elements of Z^*_N. An encryption of m is $c=m^e \bmod N$. it is easy to verify the Dec $(sk,c)=c^d \bmod N$ is a valid decryption algorithm as $m^{ed}=m \bmod N$.

To verify that multiplication of RSA ciphertexts gives the ciphertext that corresponds to the message of multiplied plaintexts we observe that[6]:

Enc (m1).Enc (m2) $=m_1^e m_2^e = (m_1 m_2)^e = Enc(m_1.m_2)$

Example:

Choose p=3 and q=11
Compute n=p*q=3*11=33
Compute to Ø(N)=(p-1)(q-1)=2*10=20
Choose e such that 1<e< Ø(N) and e and n are co=prime. Let e=7
Compute a value for d such that e.d=1 mod Ø(N). one solution is d=3
Public key is (e,N)=(7,33)
Private key is (d,N)=3,33)

Example of homomorphic property: now, let m1=2 and m2=3
$C1=m1^e \bmod N=2^7 \bmod 33=29$
$C2=m2^e \bmod N=3^7 \bmod 33=9$
C1.c2=29*9=261
By decryption (C.C2) we get: $261^3 \bmod 33=6=2*3$

More Recently, Boneh, Goh and Nissim were the first to construct a scheme capable of performing both operations at the same time – their scheme handles an arbitrary number of additions but just one multiplication. More recently, in a breakthrough work, Gentry constructed a fully homomorphic encryption scheme (FHE) capable of evaluating an arbitrary number of additions and multiplications (and thus, compute any function) on encrypted data. Aside from Gentry‟ s scheme (and a variant thereof by Smart and Vercauteren and an optimization by Stehle and Steinfeld), there are two other fully homomorphic encryption schemes.

V. APPLICATIONS OF HOMOMORPHIC ENCRYPTION

Protection of mobile agents:

One of the most interesting applications of homomorphic encryption is its use in protection of mobile agents. As we have seen in Section 3, a homomorphic encryption scheme on a special non-abelian group would lead to an algebraically homomorphic cryptosystem on the finite filed F2. Since all conventional computer architectures are based on binary strings and only require multiplication and addition, such homomorphic cryptosystems would offer the possibility to encrypt a whole program so that it is still executable. Hence, it could be used to protect mobile agents against malicious hosts by encrypting them (Sander & Tschudin, 1998a). The protection of mobile agents by homomorphic encryption can be used in twoways: (i) computing withencrypted functionsand (ii) computing with encrypted data. Computation with encrypted functions is a special case of protection of mobile agents. In such scenarios, a secret function is publicly evaluated in such a way that the function remains secret. Using homomorphic cryptosystems the encrypted function can be evaluated which guarantees its privacy. Homomorphic schemes also work on encrypted data to compute publicly while maintaining the privacy of the secret data. This can be done encrypting the data in advance and then exploiting the homomorphic property to compute with encrypted data.

Multiparty computation:

In multiparty computation schemes, several parties are interested in computing a common, public function on their inputs while keeping their individual inputs private. This problem belongs to the area ofcomputing with encrypted data. Usually in multiparty computation protocols, we have a set of E y $players whereas in computing with encrypted data scenarios E $. Furthermore, in multi-party computation protocols, the function that should be computed is publicly known, whereas in the area of computing with encrypted data it is a private input of one party.

Secret sharing scheme:

In secret sharing schemes, parties share a secret so that no individual party can reconstruct the secret form the information available to it. However, if some parties cooperate with each other, they may be able to reconstruct the secret. In this scenario, the homomorphic property implies that the composition of the shares of the secret is equivalent to the shares of the composition of the secrets.

Threshold schemes:

Both secret sharing schemes and the multiparty computation schemes are examples of threshold schemes. Threshold schemes can be implemented using homomorphic encryption techniques.

Zero-knowledge proofs:

This is a fundamental primitive of cryptographic protocols and serves as an example of a theoretical application of homomorphic cryptosystems. Zero-knowledge proofs are used to prove knowledge of some private information. For instance,

Election schemes:

In election schemes, the homomorphic property provides a tool to obtain the tally given the encrypted votes without decrypting the individual votes. Watermarking and fingerprinting schemes:Digital watermarking and fingerprinting schemes embed additional information into digital data. The homomorphic property is used to add a mark to previously encrypted data. In general, watermarks are used to identify the owner/seller of digital goods to ensure the copyright. In fingerprinting schemes, the person who buys the data should be identifiable by the merchant to ensure that data is not illegally redistributed.

Oblivious transfer:

It is an interesting cryptographic primitive. Usually in a two-party 1-out-of-2 oblivious transfer protocol, the first party sends a bit to the second party in such as way that the second party receives it with probability ½, without the first party knowing whether or not the second party received the bit. An example of such a protocol that uses the homomorphic property can be found in (Lipmaa, 2003). Commitment schemes: Commitment schemes are some fundamental cryptographic primitives. In a commitment scheme, a player makes a commitment. She is able to hoose a value from some set and commit to her choice such that she can no longer change her mind. She does not have to reveal her choice although shemay do so at some point later. Some commitment schemes can be efficiently implemented using homomorphic property.

Lottery protocols:

Usually in a cryptographic lottery, a number pointing to the winning ticket has to be jointly and randomly chosen by allparticipants. Using a homomorphic encryption scheme this can be realized as follows: Each player chooses a random number which she encrypts. Then using the homomorphic property the encryption of the sum of the random values can be efficiently computed

VI. CONCLUSION:

Homomorphic encryption provides privacy and security for data processing in unreliable environments, such as cloud and grid computing. Gentry shows the fully homomorphic feasibility and Van Dijk et al define a simple scheme. Nevertheless, both schemes are limited for processing 1 bit at time. This paper extends the DGHV scheme in order to enable operations with integers of arbitrary size without transforming the numbers into bits. We prove the correctness of our scheme by requiring total noise shorter than the half of private key. Thus, the shorter the private keys, the shorter the acceptable noise and, then, the less operations we can perform. We also conclude that multiplication is the most critical operation and the main reason for limiting circuit deep.

REFERENCES

[1.] MS. PARIN V. PATEL, MR. HITESH D. PATEL, PROF. PINAL J. PATEL A "Secure Cloud using Homomorphic Encryption Scheme" International Journal of Computer Science Research & Technology (IJCSRT)Vol. 1 Issue 1, June – 2013, IJCSRTV1IS010003.

[2.] S.Hemalatha, Dr. R.Manickachezian "Performance of Ring Based Fully Homomorphic Encryption for securing data in Cloud Computing" *International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 11, November 2014*

[3.] Craig Gentry. A fully homomorphic encryption scheme. PhD thesis,Stanford Univer- sity, 2009. crypto.stanford.edu/craig.Elsevier, 2011

[4.] M. Van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers,"EUROCRYPT 2010 , pp.24–43, 2010.

[5.] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems,"CACM , vol. 21, no. 2, pp.120–126, 1978
[6.] Vic (J.R.) Winkler, "Securing the Cloud, Cloud Computer Security, Techniques and Tactics",
[7.] R. Rivest, L. Adleman, and M. Dertouzos, "On data banks and privacy homomorphisms,"Foundations of secure computation, pp. 169–179,1978.
[8.] V. Vaikuntanathan, "Computing blindfolded: New developments in fullyhomomorphic encryption," in FOCS 2011, 2011, pp. 5–16.
[9.] C. Gentry, "Fully homomorphic encryption using ideal lattices," inSTOC2009. ACM, 2009, pp. 169–178.