Enterprise Cloud Security with Outsourcing Computation

Amrit Mukherjee[1], Aditya Sinha[2]
Consultant(Faculty)[1],Head of Department[2]

Abstract

The use of cloud computing is becoming widespread, but systematic study of its managerial implications is lacking. This paper examines cloud computing in the context of other major changes in Information Technology and explores the revolutionary transformations and challenges it brings to enterprises security . The paper also analyzes the Homographic encryption for enterprise clouds and its practical concerns regarding outsourcing computation; In this paper we have discussed the Actor model computations and its benefits for business oriented clouds and its security while considering outsourcing computing.

1. Introduction

For years the Internet has been represented on networks diagram by a Cloud symbol until 2008 when a variety of new services started to emerge that permitted computing resources to be accesed over internet termed Cloud computing. Cloud computing encompasses activities such as the use of social networking sites and other forms of interpersonal computing; however, most of the time cloud computing is concerned with accessing online software applications, data storage and processing power. Cloud computing is a way to increase the capacity or add capabilities dynamically without investing in new infrastructure, training new personnel, or licensing new software. It extends Information Technology's (IT) existing capabilities. In the last few years, cloud computing has grown from being a promising business concept to one of the fast growing segments of the IT industry. But as more and more information on individuals and companies are placed in the cloud, concerns are beginning to grow about just how safe an

environment it is. Despite of all the hype surrounding the cloud, customers are still reluctant to deploy their business in the cloud. Security issues in cloud computing has played a major role in slowing down its acceptance, in fact security ranked first as the greatest challenge issue of cloud computing as depicted in figure 1.

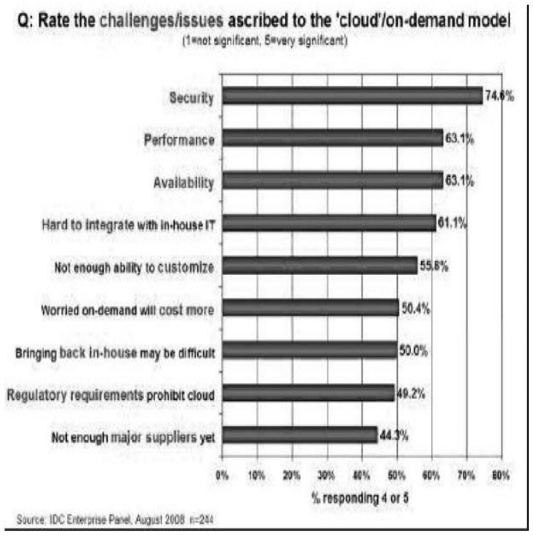

Figure 1.Results of IDC survey rankings security challenges, 2008

Cloud computing is currently characterized by having an on demand access to elastic resources via a tenancy model. It typifies the holy grail of no-worries in computing, allowing a company to focus on its core business, paying for all its IT resources as a service.

2. Cloud computing service models

In cloud computing, everything is delivered *as a Service* (*XaaS*), from testing and security, to collaboration and metamodeling [1]. The cloud was rapidly becoming a conflagration of buzzwords "as a service". Today there are three main service models, which are agreed on and defined in the NIST document [2].

Software as a Service (SaaS)

this simply means delivering software over the Internet. It is the most widely known model of cloud computing. *SaaS* has been around since early 2001 when it was

commonly referred to as the Application Service Provider (ASP) Model [8]. Software as a Service consists of software running on the provider's cloud infrastructure, delivered to (multiple) clients (ondemand) via a thin client (e.g. browser) over the Internet.Typical examples are Google Docs and Salesforce.com

Platform as a service(Paas)
this gives a client (developer) the flexibility to build (develop, test and deploy) applications on the provider's platform (API, storage and infrastructure). *PaaS* stakeholders include the *PaaS* hoster who provides the infrastructure (servers etc), the *PaaS* provider who provides the development tools and platform and the *PaaS* user [3]. Examples of *PaaS* are Microsoft Azure and Google AppEngine.

Infrastructure as a Service (IaaS)
rather than buy servers and build a datacenter from ground up, and consequently having to worry about what happens when the website hits a million users, *IaaS* offers users elastic on demand access to resources (networking, servers and storage), which could be accessed via a service API. The underlying infrastructure is transparent to the end user, while s/he retains control over the platform and software running on the infrastructure. *IaaS* runs on a tenancy model, which employs a usage-based payment approach allowing users to pay for only those resources they actually use.

3. Cloud computng Characterisics

This cloud model promotes availability and is composed of five essential **characteristics**. It is a model for enabling convenient on demand network access to a shared pool of configurable computing and outsourcing computing with respect to enterprises.

On-Demand Self service
The client has provision capabilities, such as Server time and network storage, as needed automatically without requiring manual interaction with each service's provider.

Broad network access
Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms

Resource pooling
The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines.

Rapid elasticity
Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.

Meausured Service
Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported providing transparency for both the provider and consumer of the utilized service.

4. Cloud Deployment Models

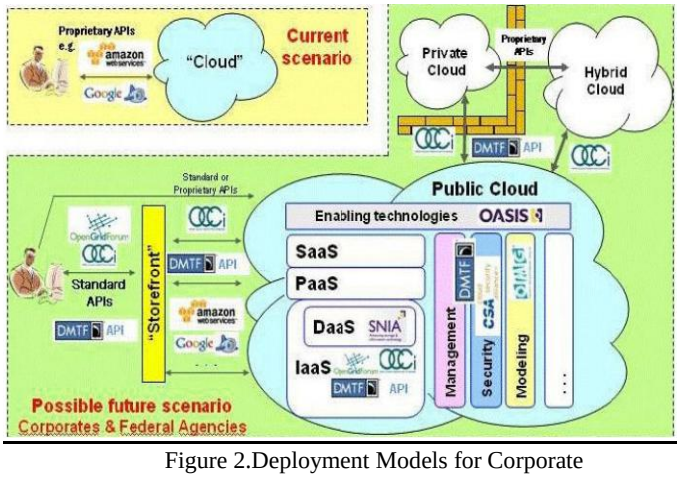

Figure 2.Deployment Models for Corporate and Federal Agencies

Private cloud
The cloud infrastructure is operated solely for an organization. It may be managed by the organization or a third party and may exist on premise or off premise.

Community cloud
The cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be managed by the organizations or a third party and may exist on premise or off premise.

Public cloud
The cloud infrastructure is made available to the general public or a large industry group and is owned by an organization selling cloud services.

Hybrid cloud
The cloud infrastructure is a composition of two or more clouds (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load-balancing between clouds).

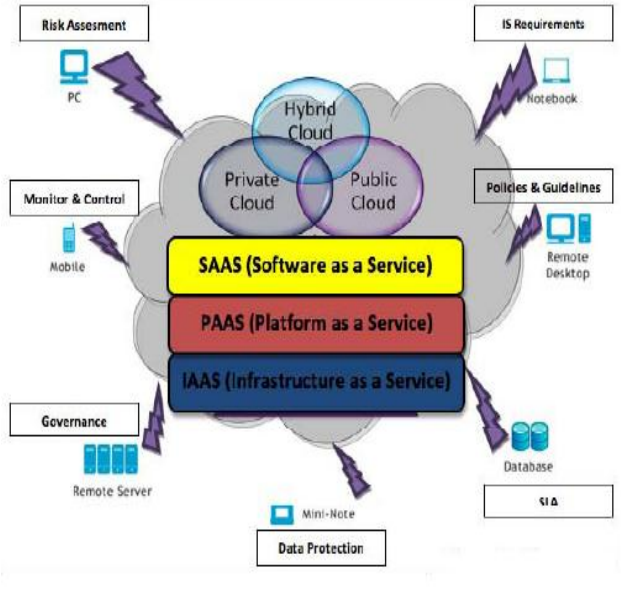

Figure 3.Cloud with all models

5. Cloud Growth in outsourcing computation
Cloud computing is a combination of several key technologies that have evolved and matured over the years. This evolution to present day cloud computing includes a combination of open API's, storage, computing, infrastructure. A service provider (such as Google or Amazon) provides hardware and virtualization software, and sometimes also applications. Instead of us, as users, hosting our own servers, their computers run virtual machines where our server can reside. The service provider's machine has a certain type of software that does virtualization (VM ware), so that a single machine in the provider can run many virtual machines, each with own operating system, hard drives, and application software. If virtualization were done only through software, it would be very slow, so chip and CPU manufacturers like INTEL and AMD embedded in their chip set hardware that allows virtualization support to create virtual machines. To avoid the security problems of virtual machines, manufacturers such as INTEL make the chip CPU with a special code so that when we, as users, log into the virtual machine in the provider's physical machine, we can be sure that the host machine is indeed ours, and not accessible to others without our permission.

The Profitability of cloud computing can be explained in the "cost associativity" formulae shown in Formula 1., the left-hand side multiplies the net revenue per userhour by the number of user-hours, giving the expected profit from using cloud computing while the right-hand side performs the same calculation for a fixed-capacity datacenter by factoring in the average utilization, including nonpeak workloads, of the datacenter; whichever side is greater represents the opportunity for higher profit" (Armbrust et al., 2009). Ambrust et al. (2009, p. 10-11) gave example on elasticity with calculations on the potentials of cloud computing savings and cost reduction:

$$UserHours_{cloud} \times (revenue - cost_{cloud}) \geq UserHours_{datacenter} \times (revenue - (cost_{datacenter}/utilization))$$

Cost minimization of cloud computing to enterprises can be explained in cloud computing elasticity capability. Assume our service has a predictable daily demand where the peak requires 500 servers at noon but the trough requires only 100 servers at midnight, as shown in the figure 4.

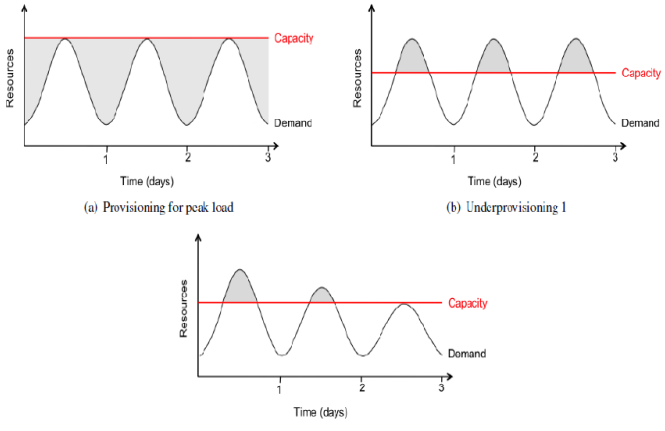

Figure 4.Provisioning for peak load and underprovisiong (Armbrust et al., 2009 p. 11)

As long as the average utilization over a whole day is 300 servers, the actual utilization over the whole day (shaded area under the curve) is 300 x 24 = 7200 server-hours; but since we must provision to the peak of 500 servers, we pay for 500 x 24 = 12000 serverhours, a factor of 1.7 more than what is needed. Therefore, as long as the pay-as-you-go cost per server-hour over 3 years is less than 1.7 times the cost of buying the server, we can save money using utility computing (Armbrust et al., 2009 p. 10).

In Figure 4,we observed: (a) Even if peak load can be correctly anticipated, without elasticity we waste resources (shaded area) during nonpeak times. (b) Underprovisioning case 1: potential revenue from users not served (shaded area) is sacrificed. (c) Underprovisioning case 2: some users desert the site permanently after experiencing poor service; this attrition and possible negative press result in a permanent loss of a portion of the revenue stream (Armbrust et al., 2009).

6. Enterprise security challenges

Cloud computing faces just as much security threats that are currently found in the existing computing platforms, networks, intranets, internets in enterprises. These threats, risk vulnerabilities come in various forms. The Cloud Security Alliance (Cloud Computing Alliance, 2010) did a research on the threats facing cloud computing and it identified the following seven major threats:

- Abuse and Nefarious Use of Cloud Computing
- Insecure Application Programming Interfaces
- Malicious Insiders
- Shared Technology Vulnerabilities
- Data Loss/Leakage
- Account, Service & Traffic Hijacking
- Unknown Risk Profile

Moving to the cloud presents the enterprise with a number of risks and that include securing critical information like the protection of intellectual property, trade secrets, personally identifiable information that could fall into the wrong hands. Making sensitive information available on the internet requires a considerable investment in security controls and monitoring of access to the contents. In the cloud environment, the enterprise may have little or no visibility to storage and backup processes and little or no physical access to storage devices by the cloud computing provider. And, because the data from multiple customers may be stored in a single repository, forensic inspection of the storage media and a proper understanding of file access and deletion will be a significant challenge (Information Security Magazine, 2009).

The researchers went on to say that a way around the weakness they found in Amazon's EC2 is for customers to insist that their cloud machines are placed on physical machines that only they can access or that they and trusted third parties can access (Greene, 2009). This solution will likely be at a price premium because part of the economy of cloud services is maximizing use of physical servers by efficiently loading them up with cloud machines (Greene, 2009) and locating the cloud datacenter where the utility price is the cheapest.

Because of privacy concerns, enterprises running clouds collecting data have felt increasing pressure to anonymize their data. EPIC has called for Gmail, Google Docs, Google Calendar, and the company's other Web applications to be shut down until appropriate privacy guards are in place [7]. Google and

Yahoo!, because of pressure from privacy advocates, now have an 18 month retention policy for their search data, after which it will be anonymized. This means that some identifying data will be removed such as IP addresses and cookie information. The anonymized data is retained though, to support the continual testing of their algorithms. Another reason to anonymize data is to share data with other parties. These may be to support research (e.g., the AOL incident [6]) or to subcontract out data mining on the data (e.g., the Netflix data set [6]).

An example of indirect data-mining that might be performed by a cloud provider is to note transactional and relationship information (see World Privacy Forum Report [5]). For example, the sharing of information by two companies may signal a merger is under consideration.

Availability also needs to be considered in the context of an adversary whose goals are simply to sabotage activities. Increasingly, such adversaries are becoming realistic as political conflict is taken onto the web, and as the recent cyber attacks on Lithuania confirm [6]. The damages are not only related to the losses of productivity, but extend to losses due to the degraded trust in the infrastructure, and potentially costly backup measures. The cloud computing model encourages single points of failure. It is therefore important to develop methods for sustained availability (in the context of attack), and for recovery from attack. The latter could operate on the basis of minimization of losses, required service levels, or similar measures.

The development of cloud computing may, in the extreme, allow the use of thin clients on the client side. Rather than a license purchased and software installation on the client side, users will authenticate in order to be able to use a cloud application. There are some advantages in such a model, such as making software piracy more difficult and giving the ability to centralize monitoring. The development has potential security implications, both in terms of data leaks, and in terms of the number of sources of data a user may have to pull data from – this, in turn, places requirements on how access is authorized for reasons of usability. While centralized access control may solve many of these problems, that may not be possible – or even desirable.

One example in this area is provided by Facebook. Facebook users upload both sensitive and non-sensitive data. This data is both utilized by Facebook to present the data to other users, and also utilized by third party applications that are run by the platform. These applications are typically not verified by Facebook. Hence, there is a drive to create malicious applications that run in Facebook's cloud to steal sensitive data, e.g., see [4]. Furthermore, Abadi pointed out that it is hard to maintain ACID (atomicity, consistency, isolation, durability) properties of during data replication over large geographic zones. Data remembrance or persistence remains an issue due to replication and distribution of data even after a user has left a cloud provider.

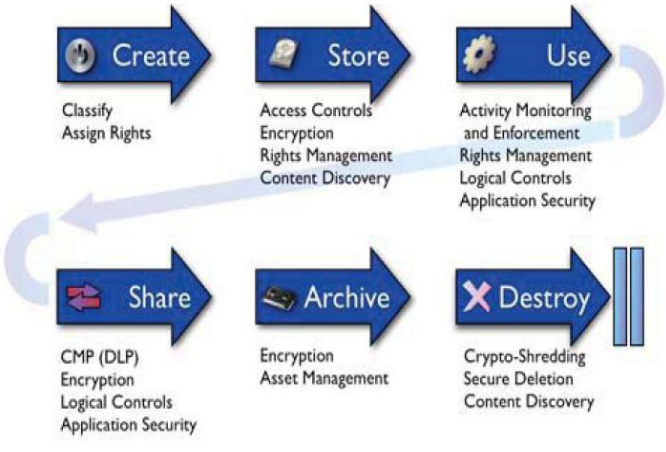

Figure 5.Data security life cycle

Data in the cloud is usually globally distributed which raises concerns about jurisdiction, data exposure and privacy. Pearson [7] summarized the main privacy issues of cloud computing. Users are made to give away their personal information without knowing where it is stored or what future purpose it might serve. Organizations stand a risk of not complying with government policies as would be explained further while the cloud vendors who expose sensitive information risk legal liability. Virtual co-tenancy of sensitive and non-sensitive data on the same host also carries its own potential risks.

7. Aplying homomorphic in outsourcing computation

What if the user could carry out any arbitrary computation on the hosted data without the cloud

provider learning about the user's data - computation is done on encrypted data without prior decryption. This is the promise of homomorphic encryption schemes which allow the transformation of ciphertexts *C(m)* of message m, to ciphertexts *C(f(m))* of a computation/function of message m, without disclosing the message. The idea was first suggested by Rivest, Adleman and Dertouzos in 1978, referred to as privacy homomorphisms [13]. RSA (invented by Rivest, Shamir and Adleman '78) [15] had multiplicative homomorphism (you could compute a ciphertext which is the product of plaintexts) and over the next 30 years, researchers such as A.Shamir [15],ElGamal '85 [14] and Paillier [13] came up with partially homomorphic cryptosystems. A survey of homomorphic encryption schemes can be found in [13,15].

An encryption scheme can be said to be fully homomorphic if:

$$E(m1 \Theta m2) \leftarrow E(m1)\ \Theta\ E(m2); \text{for all } m1, m2 \in M \quad (1)$$

Where M is the set of plaintexts, **Θ** - represents any arbitrary function and ← means computation is done without the plaintexts being decrypted. The first fully homomorphic encryption system was proposed by Craig Gentry in 2009 [16] using ideal lattices. Gentry's approach employed devising a somewhat homomorphic scheme, and then bootstrapping it to get a fully homomorphic scheme. Since then researchers have proposed variants and improvements to Gentry's model.

Smart and Vercauteren [19] presented a specialization of Gentry's scheme which yielded a smaller ciphertext size. Dijk, Gentry, Halevi, and Vaikuntanathan [17] introduced the first variant of Gentry's using arithmetic operations over integers. Stehle and Steinfield [18] also proposed a faster improvement of Gentry's model.

GENTRY'S fully Homomorphic Encryption using ideal lattices:

An encryption scheme χ has the following three step algorithm:

1. ***KeyGen*** χ - creates two keys i.e. the secret key sk and the public key pk.

2. ***Encrypt*** χ - encrypts the plaintext m with the public key pk to yield ciphertext c.

3. **Decrypt** χ - decrypts the ciphertext c with the secret key sk to retrieve the plaintext m.

Gentry introduced a fourth step called *Eval* to the algorithm:

4. ***Eval*** χ - outputs a ciphertext c of f(m) such that

Decrypt χ *(sk,m) = f(m).*

The scheme becomes homomorphic if *f* can be any arbitrary function, and the resulting ciphertext of *Eval* is compact (i.e. it does not grow too large regardless of the complexity of function *f*). The *Eval* algorithm in essence means that the scheme can evaluate its own decryption algorithm (i.e. the scheme is *bootstrappable*).

It can be shown that any arbitrary function is made up of an aggregate of addition, subtraction and multiplication functions (i.e. AND, OR and NOT gates). Gentry employed ideal lattices which provide additive and multiplicative homomorphism and low circuit complexities (for the decryption algorithm) in creating his fully homomorphic scheme. More on lattice based cryptography can be found in [5,6]. Further, in order to prevent the ciphertext (as well as the inherent noise/error) from becoming too large, Gentry introduced a *Recrypt* χ function which refreshes a ciphertext *c'* to produce a new ciphertext *c* using a different key. *Recrypt* χ is a two-step procedure:

1. *Encrypt* χ *(pk2, c1)* to yield *c1'*, then

2. output a new ciphertext *c* by *Eval* χ *(pk2,*

Decrypt χ *, sk2', c1')*.

This is the process behind bootstrapping to yield the fully homomorphic encryption. The main issue with this scheme above is the ciphertext and noise growth (especially w.r.t. function complexity).

8. Implementing actor model computation for enterprise

An Actor is a computational entity that, in response to a message it receives, can concurrently:

- send messages to other Actors;

- create new Actors;
- designate how to handle the next message it receives.

There is no assumed order to the above actions and they could be carried out concurrently. In addition two messages sent concurrently can arrive in either order. Decoupling the sender from the communications it sends was a fundamental advance of the Actor model enabling asynchronous communication and control structures as patterns of passing messages [Hewitt 1977].

An Actor can only communicate another Actor to which it has an address. Addresses can be implemented in a variety of ways:

- direct physical attachment
- memory or disk addresses
- network addresses
- email addresses

The Actor Model differs from its predecessors and most current models of computation in that the Actor model assumes the following:

- Concurrent execution in processing a message.
- The following are not required by an Actor: a thread, a mailbox, a message queue, its own operating system process, etc.
- Message passing has the same overhead as looping and procedure calling.

Implementing Actor model can give intelligence for the cloud where the information is stored or to be act upon some operation like EC2 (Elastic Cloud Computing) and for the service. Here the noise which was earlier with Homomorphic encryption can be avoided as the Actors acts on each other rather than acting directly with the information.

Computational Representational Theorem: The Computational Representation Theorem [Clinger 1981; Hewitt 2006] characterizes computation for systems which are closed in the sense that they do not receive communications from outside. The denotation Denote_S of a closed system **S** represents all the possible behaviors of **S** as

$$\textbf{Denote}_S = \textbf{Limit}_{i\to\infty}\ \textbf{Progression}_s^{\,i} \quad (2)$$

where Progression_S takes a set of partial behaviors to their next stage, i.e., $\text{Progression}_S^{\,i} \to^4 \text{Progression}_s^{\,i+1}$ .In this way, S can be mathematically characterized in terms of all its possible behaviors (including those involving unbounded nondeterminism).Acting the Actor models in the source and in the packets which are communicating can be secured by multi-stage progressions.

9. Conclusion

Cloud is in warning from the perceived loss of control of sensitive data. Current control measures do not adequately address cloud computing's third-party data storage and processing needs. In our vision, we propose to extend control measures from the enterprise into the cloud through the use of Trusted Computing and applied cryptographic techniques and by computational theorem. These measures should alleviate much of today's fear of cloud computing, and we can, have the potential to provide demonstrable business intelligence advantages to cloud participation. Although and always, Enterprise should verify and understand cloud security, carefully analyze the security issues involved and plan for ways to resolve it before implementing the technology. For more security, Pilot projects should be setup and good governance should be put in place to effectively deal with security issues and concerns. Namely, the new threats require new constructions to maintain and improve security. Among these are tools to control and understand privacy leaks, perform authentication, and guarantee availability in the face of cloud denial-of-service attacks. Encryption in Homomorphic is although suitable for large informational clouds (Enterprise) where information intelligence is highly required for safer connections and usage.

10. References

[1] Armbrust,M.,Fox,A.,Griffith,R.,Joseph,A.D.,Katz,R,Konwinski,A.,Lee,F.,Patterson,D.,Rabkin,A.,Stoica,I.&Zaharia,M." Above the clouds :A Berkeley view of cloud computin."*Uniersity of California,Berkeley,Reliable Adaptive Distributed Systems Laboratory,Technical Report No. UCB/EECS-2009-28,(2009)*

[2] Cagle,K.,"But what exacltly 'is' cloud computing?"*O'Rielly Broadcast*,2008.

[3] Al Bento,Regina Bento,"Cloud Computing:A new phase in Information Technology" University of

Baltimore,*Journal of Information Technology ManagementVolume XXII,Number1,2011*,pp.-39-46

[4] Richard Chow,Philippe Golle,Markus Jacobsson,Ryusuke Masuoka,Jesus Molina,"Controlling data in the Cloud:Outsourcing Computation without Outsourcing control"

[5] Aderemi A. Atayero,Oluwaseyi Feyisetan,Covenant Universty,Nigeria,Kings College,United Kingdom,"Securty Isues in Cloud Computing :The Potentials of Homomorphic Encryption",*Journal of Emerging Trends in Computing and Information Sciences,Vol 2,No-10,October 2011*,pp.546-552

[6] Anthony Bisong and Syed(Shawon) M.Rahman,Capella University,Minneapolis,"An overview of the security concerns in Enterprise Cloud Computing",*International Journalof Network Security & its applications(IJNSA),Vol.-3,No-1,January 2011*,pp.30-45

[7] KuyoroS.O.,Ibikunle F.,Awodele O.,Dept. of Computer Science,Babcock University,Covenant University,Nigeria,*International Journal of Computer Networks(IJCN),Volume(3):Issue(5):2011*,pp.247-255

[8] Sara Qaisar,Kausar Fiaz Khawaja,Dept. of Technology Management,Faculty of Management Sciences,Islamabad,Pakistan,"Cloud Computing:Network/Security Threats and coutermeasures",*Interdisciplinary Journal of Cotemporary Research in Business,Vol-3,No-9,January 2012*,pp.1323-1329

[9] Ajith Singh. N,M.Hemalatha,Dept. of Computer Science,Dept. of Software Systems and Research,Karpagam University,Coimbatore,India,"Cloud Computing for Academic Environment",*International Journal of Information and Communication Technology Research.,Vol.2,No-2,February-2012*,pp.97-101

[10] Rittinghouse,J.W., & Ransome,J.F. (2010). Cloud Computing Implementation, Management, and Security. New York: Taylor and Francis Group.

[11] Khmelevsky,Y., and Voytenko,V. (2010).Cloud Computing Infrastructure Prototype for University Education and Research. Proceedings of the 15th Western Canadian Conference on Computing Education. Kelowna, Canada: ACM.

[12] The Research and Application of Network Teaching Platform Based on Cloud Computing, Zhang Tao and Jiao Long, International Journal of Information and Education Technology, Vol. 1, No. 3, August 2011.

[13] R.Rivest,L.Adleman and M.Dertouzos.On data bank and privacy homomorphisms.In Foundation of secure computation.,pp. 169-180,1978.

[14] S.Goldwasser,S.Micali,"Probabilistic encryption",Journal of Computer and Sytem Sciences,vol-28.,pp 270-299,1984

[15] R.Rivest,A.Shamir and L.Adleman.A method for obtaining digital signatures and public key crptosystems.In Comm. Of the ACM 21:2,pp. 120-126,1978.

[16] C.Gentry .Fully Homomorphic Encryption using ideal lattices.In proc. of STOC,pp. 169178.ACM,2009.

[17] M.van Dijk, C.Gnetry,S.Halevi and V.Vaikuntham,Fully Homomorphic Encryption over Integers.In Advances in Cryptology-Eurocrypt 2010,Springer LNCS 6110,pp. 24-43,2010.

[18] Goh.,E.J.:Secure Indexes Technical Report 2003/216,IACR eprint Cryptography Archieve(2003)

[19] N.P.Smart and F.Vercauteren.Fully Homomorphic Encryption with relatively Small Key and Cyphertext sizes.Lecture notes in Computer Science,2010 Volume 6056/2012,pp. 420-443,2010