International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 3, Issue 1, July 2014)

A Survey of Homomorphic Encryption Schemes in Cloud Data Storage

R. Kanagavalli[1], Dr. Vagdevi S[2]

[1]*Associate Professor, Department of CSE, Global Academy of Technology, Bangalore-560098. Karnataka, India.*
[2] *Professor & HOD, Department of ISE, Global Academy of Technology,Bangalore-560098. Karnataka, India*

***Abstract*— Cloud Computing technology is enabling IT managers to treat infrastructure as a common substrate on which they can provide services to users in a faster, more flexible and cost effective way. Despite the trumpeted business and technical advantages of cloud computing, many potential users are yet to join the cloud and those who are cloud users put only their less sensitive data in the cloud. Lack of security in the cloud is a major worry. Data can be encrypted and stored in the cloud but the problem is that while data can be sent to and from a cloud provider's data center in an encrypted form, the servers that power a cloud can't do any work on it that way. With homomorphic encryption, a company could encrypt its entire database and upload it to a cloud and it is possible to analyze data without decrypting it. In this paper, we surveyed homomorphic cryptosystems, how they differ, how to use them to secure our data in cloud and perform operations on the data with the help of cipher texts**.

***Keywords*— Cloud Security, Cloud Data Storage, Homomorphic key, Erasure coding, RNS.**

I. INTRODUCTION

Cloud computing is a new technique put forward from the industry circle. It encompasses parallel computing, distributed computing and grid computing. It is a combination and evolution of virtualization, utility computing, Infrastructure-as-a-service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). Cloud computing can provide three kinds of service modes. SaaS refers to the services provided to clients, the applications running on the cloud computing infrastructure provided by the service providers. PaaS refers to deploying the applications created by the development language and tool provided by the service providers to the cloud infrastructure. IaaS refers to the services provided to the users, to lease the processing power, storage network and other basic computing resources. For all these services, there is no need for users to manage or control the cloud infrastructure, including network, server, operating system, storage and even the functions of applications.

Cloud computing is an internet based development and use of computer technology and computing architecture in which data centers are transformed into pools of computing services by powerful processors together with SaaS architecture. Cloud computing is a result of usage of technology for day to day activities through internet Cloud computing came into the foreground as there were advances in virtualization, distributed computing with server clusters and increase in the availability of broadband internet access. IT industry describes cloud computing simply as the delivery of applications provided by a third party over the internet. Some of the service providers are Rack space, Microsoft, and IBM. The buzz began way back in 2006 with the launch of Amazon EC2, gained traction in 2007 and currently characterized by having an on demand access to elastic sources through a tenancy model.

The remainder of the paper is organized as follows. In Section II, we introduce Homomorphic Encryption and its usage in cloud data storage. In Section III, we present our survey on existing work in the field. In Section IV, we present our proposed work. In Section V, we present our survey results and the conclusion of the survey is mentioned in Section VI.

II. HOMOMORPHIC ENCRYPTION

If all data stored in the cloud is in the encrypted form that would effectively solve issues like Availability, Data Security, and third party control. But a user would not be able to trust the power of the cloud to carryout computation of data without first decrypting it or shipping it entirely back to the user for computation. So, the cloud provider has to decrypt the data first thus nullifying the issues of privacy and confidentiality, perform the computation and then send the result to the user. Suppose if the user could carry out any arbitrary computation on the hosted data, then without the cloud provider learning about the user's data, computation is done on encrypted data without prior decryption.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 3, Issue 1, July 2014)

In this scenario, the promise of homomorphic encryption takes a call that homomorphic encryption schemes which allow the transformation of cipher texts *C(m)* of message m, to cipher texts *C(f(m))* of a computation/function of message m, without disclosing the message. The first homomorphic encryption scheme was suggested by Rivest, Alderman and Dertouzos [1] in 1978 popularly known as privacy homomorphism. An encryption scheme is said to be fully homomorphic if from $E_n(a)$ and $E_n(b)$,it is possible to compute $E_n(f(a,b))$ where f can be +, x ,⊕ and without using the private key. The homomorphic encryption schemes can be classified according to the operation that allows to assess on raw data, as the additive homomorphic encryption and multiplicative homomorphic encryption. Additive homomorphic encryption schemes are schemes in which the cipher texts are computed as the sum of plain texts. Additive homomorphic encryption schemes are Pailler cryptosystem [2] and Goldwasser –Micalli [3] cryptosystems.

Key Generation: KeyGen(p,q)	
Input: $p,q \in \mathbb{P}$	
Compute	$n = pq$
Choose $g \in \mathbb{Z}_{n^2}^*$ such that	$\gcd(L(g^\lambda \bmod n^2), n) = 1$ with $L(u) = \frac{u-1}{n}$
Output: (pk, sk)	
public key: $pk = (n, g)$	
secret key: $sk = (p, q)$	

Encryption: Enc(m, pk)	
Input: $m \in \mathbb{Z}_n$	
Choose	$r \in \mathbb{Z}_n^*$
Compute	$c = g^m \cdot r^n \bmod n^2$
Output: $c \in \mathbb{Z}_{n^2}$	

Decryption: Dec(c, sk)	
Input: $c \in \mathbb{Z}_{n^2}$	
Compute	$m = \frac{L(c^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n$
Output: $m \in \mathbb{Z}_n$	

Fig 1 Pailler Algorithm

Multiplicative homomorphic encryption schemes are schemes in which cipher texts are calculated as the product of plain texts. Multiplicative homomorphic schemes are RSA and ElGamal cryptosystems. For all types of calculations on the data stored in cloud, Fully Homomorphic Encryption (FHE) must be opted because FHE is able to execute all types of operations on encrypted data without decryption. The application of FHE is an important stone in cloud computing security.

Using FHE, we could outsource the calculations on confidential data to cloud server keeping the secret key that can decrypt the result of calculation. In our work we encrypt the data before sending it to the cloud service provider using FHE schemes

Key Generation: KeyGen(p,q)	
Input: $p,q \in \mathbb{P}$	
Compute	$n = p \cdot q$
	$\varphi(n) = (p-1)(q-1)$
Choose e such that	$\gcd(e, \varphi(n)) = 1$
Determine d such that	$e \cdot d \equiv 1 \bmod \varphi(n)$
Output: (pk, sk)	
public key: $pk = (e, n)$	
secret key: $sk = (d)$	

Encryption: Enc(m, pk)	
Input: $m \in \mathbb{Z}_n$	
Compute	$c = m^e \bmod n$
Output: $c \in \mathbb{Z}_n$	

Decryption: Dec(c, sk)	
Input: $c \in \mathbb{Z}_n$	
Compute	$m = c^d \bmod n$
Output: $m \in \mathbb{Z}_n$	

Fig.2 RSA Algorithm

A cloud storage system is considered as a large scale distributed storage system that consists of many independent storage servers. Data robustness is a major requirement storage systems. One way to provide data robustness is to replicate a message such that each storage server stores a copy of the message. This method ensures robustness because the message can be retrieved as long as one storage server survives. However the method is expensive as z replicas result in z times of expansion. One way to reduce the expansion rate is to use erasure codes to encode messages. A message is encoded as a codeword, which is a vector of symbols and each storage server stores a codeword symbol. A storage server failure is modeled as an erasure error of the stored codeword symbol. Random linear codes support distributed encoding, i.e., each codeword symbol is independently computed. To store a message of k blocks, each storage server linearly combines the blocks with randomly chosen coefficients and stores the codeword symbol and coefficients.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 3, Issue 1, July 2014)

To retrieve the message, a user queries k storage servers for the stored codeword symbols and coefficients and solves the linear systems. Our survey study is done by considering a cloud storage system in which there *n* distributed storage servers and *m* key servers. A message is divided into *k* blocks and represented as a vector of *k* symbols.

III. Related Work

Dimakis et al. [4] considered the case that $n= ak$ for a fixed constant *a*. It is shown that distributing each block of a message to *v* randomly chosen storage servers is enough to have a probability *1-k/p-o(1)* of successful data retrieval ,where $v=b \ln k$,$b>5a$ and *p* is the order of used group. The sparsity parameter $v = b \ln k$ is the number of storage servers which a block is sent to. The larger *v* is, the communication cost is higher and the successful retrieval probability is higher. The system has a light confidentiality because an attacker can compromise *k* storage servers to get the message.

Lin and Tzeng [5] addressed robustness and confidentiality issues by presenting a secure decentralized erasure code for the networked storage system. In addition to storage servers, their system consists of key servers, which hold cryptographic key shares and work in a distributed way. In their system, stored messages are encrypted and encoded. To retrieve a message, key servers query storage servers for the user. As long as the number of available key servers is over a threshold t, the message can be successfully retrieved with an overwhelming probability. Their results shows that when there are *n* storage servers with $n= ak\sqrt{k}$, the parameter *v* is $b\sqrt{k} \ln k$ with $b>5a$ and each key server queries 2 storage servers for each retrieval request, the probability of a successful retrieval is at least *1-k/p-o(1.)*

Hsiao-Ying Lin and Wen-Guey Tzeng [6] developed a highly distributed system with parameters $n=a\ k^c$ where $c\geq 1.5$ *and* $a > \sqrt{2}$. The system supports secure data forwarding by using threshold proxy re-encryption scheme. Here the number of storage servers is much greater than the number of blocks of a message.

Mambo and Okamato [7] and Blaze et al.[8] proposed proxy re-encryption schemes. In a proxy re-encryption scheme, a proxy server can transfer a cipher text under a public key PK_A to a new one under another public key PK_B by using the re-encryption key $RK_{A\text{-}B}$. The server does not know the plain text during transformation

Ateniese et al. [9] proposed a system in which messages are first encrypted by the owner and then stored in a storage server. When a user wants to share his messages, he sends a re-encryption key to the storage server. The storage server re-encrypts the encrypted messages for the authorized user .

Mahadevan Gomathikrishnan et al.[10] proposed a homomorphic encryption scheme using Residue Number System[RNS]. Here a program *P* is threaded into *k* threads. All the key variables to be protected are split into *k* moduli using RNS. Each thread would have identical control flow different data. In this scheme a secret is split into multiple shares on which computation can be performed independently. Security is enhanced by not allowing independent clouds to collude. Efficiency is achieved through the smaller shares.

Emalda roslin et al. [11] proposed a method in which a cloud server splits into different chunks and then encrypted. Then the encrypted cloud server is kept in a replica cloud server as a backup. The encrypted data is converted into bytes and then added with parity bit by the data owner in order to restrict the TPA by accessing the original data.

Osama Khan et al. [12] proposed an algorithm which finds the optimal number of code word symbols needed for recovery for any XOR based erasure code and produces recovery scheduler that use a minimum amount of data.

Aderemi and Oluwaseyi [13] proposed a searchable symmetric encryption scheme which permits a user to selectively search the data that the user hosted in the cloud. An encryption layer on the top of the encrypted files to be stored in the cloud. This extra layer would be an encrypted search index layer which can be searched using secure indexes.

Cloud computing security is an evolving sub-domain of computer security, network security and more broadly information security.

IV. Proposed Work

In cloud data storage, a user stores his data through Cloud Service Provider (CSP) into a set of cloud servers. For application purposes the user interacts with the cloud servers via the service provider. Security threats faced by cloud data storage can come from two different sources. 1. A cloud service provider can be self interested, untrusted and possibly malicious. 2. There may also exist a combatant who compromises a number of cloud data storage servers while remaining undetected by CSP. Our model is proposed to be built by considering the second type of attack. There are two types of combatants,

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 3, Issue 1, July 2014)

1. *A combatant is interested in corrupting users data files stored in individual servers.*
2. *The combatants can compromise all the storage servers so as to modify the data files as long as they are internally consistent.*

Our model will be built by considering both the types of combatants. Our work aims at providing cloud data storage security with dynamic data support. The data distribution across the cloud servers will be done with the help of homomorphic token and distribution of files using erasure correcting codes in particular with Reed Solomon correcting codes.

V. Results Of Comparison

The existing literature works were studied in the simulation environment using CloudSim for various parameters. The parameters studied are 1.Confidentiality 2.Robustness 3. Cost of Computation 4. Efficiency 5. Dynamic Data Support and the comparison results are as shown in the below table.

TABLE I
COMPARISON RESULTS

Sl.No	Method	Parameters				
		1	2	3	4	5
1	Distributed Servers	Low	Low	High	Less	No
2	Decentralized Erasure Code	High	High	High	Less	No
3	Re-encryption	Low	Low	High	Medium	No
4	Proxy Re-encryption	High	High	High	Medium	No
5	Threshold proxy re-encryption	High	High	High	High	No
6	Chunk encryption and replica	High	Low	Less	Medium	No
7	Homomorphic encryption using RNS	High	High	Less	High	No
8	XOR based erasure codes	High	Low	Less	High	No
9	Searchable indexes	High	High	Less	High	No

VI. Conclusion

The cloud Computing Security based on Homomorphic Encryption is a new concept of Security. In this paper, we investigated the problem of data security in cloud data storage and its requirement for homomorphic encryption schemes. We analysed the existing work and have found that there are possible directions of future research in the domain of cloud data storage security. We believe that data storage security in cloud computing is an area full of challenges and of paramount importance. We propose a system with homomorphic encryption and erasure coding which is still in its infancy stage. As a result of our survey, we envision several possible directions for future research on this area. We also have planned to investigate the problem of fine grained data error localization in our future work.

REFERENCES

[1] Ronald L. Rivest, Leonard Adleman, and Michael L. Dertouzos. On Data Banks and Privacy Homomorphisms, chapter On Data Banks and Privacy Homomorphisms, pages 169-180. Academic Press, 1978.

[2] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In 18th Annual Eurocrypt Conference (EUROCRYPT'99), Prague, Czech Republic, volume 1592, 1999.

[3] Julien Bringe and al. An Application of the Goldwasser-Micali Cryptosystem to Biometric Authentication, Springer-Verlag, 2007.

[4] A.G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Decentralized Erasure Codes for Distributed Networked Storage," IEEE Trans. Information Theory, vol. 52, no. 6 pp. 2809-2816, June 2006.

[5] H.-Y. Lin and W.-G. Tzeng, "A Secure Decentralized Erasure Code for Distributed Network Storage , ," IEEE Trans.Parallel and Distributed Systems, vol. 21, no. 11, pp. 1586-1594, Nov. 2010.

[6] H.-Y. Lin and W.-G. Tzeng, "A Secure Erasure Code based cloud storage system with secure data forwarding," IEEE Trans. Parallel and Distributed Systems, vol. 23, no. 6, pp. 995-1003, June. 2012.

[7] M. Mambo and E. Okamoto, "Proxy Cryptosystems: Delegation of the Power to Decrypt Ciphertexts," IEICE Trans. Fundamentals of Electronics, Comm. and Computer Sciences, vol. E80-A, no. 1, pp. 54- 63, 1997.

[8] M. Blaze, G. Bleumer, and M. Strauss, "Divertible Protocols and Atomic Proxy Cryptography," Proc. Int'l Conf. Theory and Application of Cryptographic Techniques (EUROCRYPT), pp. 127-144, 1998.

[9] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage," ACM Trans. Information and System Security, vol. 9, no. 1, pp. 1-30, 2006.

[10] Mahadevan Gomathikrishnan,Akilesh Tyagi," HORNS-A Homomorphic encryption scheme for cloud computing using Residue Number System", IEEE transactions on parallel and distributed systems, vol. 23, no. 6, June 2011,pp 995-1003.

[11] Emalda Roslin, Abhirami, Nandita,"SSS-ECSecure storage services and erasure code implementation in cloud computing ", IJETT, vol.4, no.3, June 2013,pp 275-283.

[12] O. Khan, R. Burns, J. Plank, W. Pierce, and C. Huang, "Rethinking erasure codes for cloud file systems: Minimizing i/o for recovery and degraded reads, FAST 2012.

[13] Aderemi A Atayeo,Oluwaseyi Feyistan,"Security Issues in Cloud Computing-The potentials of Homomorphic Encryption",JETCI,Vol 2,no 10,October 2011,pp546-552.

[14] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring Data Storage Security in Cloud Computing," Proc. 17th Int'l Workshop Quality of Service (IWQoS '09), pp. 1-9, July 2009

[15] M. Creeger, "Cloud Computing: An Overview," Queue, vol. 7, no. 5, pp. 2:3-2:4, June 2009.