Homomorphic Encryption for Data Security in Cloud Computing

Kamal Kumar Chauhan[1], Amit K.S. Sanger[2], Ajai Verma[3]
Department of Computer Science & Engineering
Meerut Institute of Technology, Meerut, India
kamalchauhans@gmail.com[1], sanger.amit@gmail.com[2], ajaiverma2010@gmail.com[3]

Abstract: **Cloud computing is recent emerging Internet based technology in IT industry. With rapid development of cloud computing over Internet, security became major issue that garnered attention of researchers from academic as well as industry. Data Security is a great barrier in adaptation of cloud computing. Traditional standard encryption methods provide security to data in storage state and transmission state. But in processing state, performing operations on data require decryption of data. At this state data is available to cloud provider. Hence traditional encryption methods are not sufficient to secure data completely.**

In this paper, we discussed homomorphic encryption methods and their applications in cloud computing to secure data in processing state. Homomorphic encryption allows user to operate encrypted data directly without decryption.

Keyword: Cloud Computing; Data Security; Cryptography; Homomorphic Encryption.

I. INTRODUCTION

Cloud computing is one of the most emerged Internet based technology that garnered a great attention of researchers from academic and industry. Cloud computing provides on-demand Services over Networks (SoN) i.e. "Access services/resources anytime from anywhere in pay-per-use fashion". Cloud computing has become fast growing segment in last few years that increased the capability of IT services without investing in new infrastructure. The name cloud computing was inspired by cloud symbol used to represent Internet in diagrams [1]. Cloud computing delivers services/resources as "X as a Service (XaaS)" to customers over network.

Major services provided by cloud computing are:

Platform as a Service (PaaS): In PaaS service provider provides platform for the creation of software that are delivered to customers over the web. PaaS allow users to create applications easily without the complexity of buying and maintaining the software and infrastructure.

Software as a Service (SaaS): With SaaS, cloud provider provides licensed application to customers as a service. Applications are connected to customers' computer via Internet and applications are owned and operated by customers. Google Apps, Salesforce are example of SaaS cloud services.

Infrastructure as a Service (IaaS): IaaS delivers the computing infrastructure such as storage server space, servers, and network equipments as on demand service. Instead of purchasing hardware from outside, users can purchase IaaS based on pay per use.

Number of companies such as Google, Amazon, Microsoft etc. are developing cloud infrastructure providing services to customers through network. Cloud computing enhanced the utilization of computing resources by sharing in number of users, which is so called utility computing.

Cloud computing is developed in four deployment model namely:

Private cloud also called internal cloud. Private cloud is operated for an individual organization. Such infrastructure is accessed only by the members of the organization. Private cloud is managed by the organization or a third party such as Amazon's Elastic Compute Cloud (EC2) or Simple Storage Service (S3).

Public cloud is made available to the public or number of industry group. In public cloud, service provider makes resources available as a service to public over a network. Customers are connected to the public cloud through an Internet. Resources may be available freely or offered on a pay-per-usage model.

Community cloud is a service model that lies somewhat between private and public clouds. A community cloud is shared among several organizations. Community cloud is managed by all the participating organizations or by a third party.

Hybrid cloud includes two or more clouds (public, private or community). In hybrid cloud service provider provides and manages some services in-house and other are provided externally.

Due to characteristics of cloud computing such as multi-tenancy, virtualization, scalability, Internet-based etc, number of security threads/challenges are found in cloud computing service delivery model[2][3][4]. These security issues are great barrier to deploy cloud computing environment. We identified data security as one of the major security issue in adaptation of cloud by users.

Main focus of us, in this paper, is discussing data security

978-1-5090-0487-4/15 $31.00 © 2015 IEEE
DOI 10.1109/ICIT.2015.39

in processing and techniques to make data secure. Standard encryption techniques used for encryption require decryption to perform operations on data. Decrypting data at cloud data server makes data available to cloud provider. Therefore, we discussed and analyzed the concept of homomorphic encryption to encrypt data. Homomorphic encryption allows users to operate directly on encrypted data.

Rest of the paper is organized as section-II gives detail description of data security in cloud computing as problem statement. In section-III, we have described some homomorphic encryption methods. Homomorphic encryption for data security in cloud and its analysis is discussed in section-IV. Finally, section-V concludes the paper.

II. PROBLEM STATEMENT: DATA SECURITY

Numbers of security threats are identified, which are barriers in adaptation of cloud computing. We found that data security is one of the major barriers for end user to adapt cloud architecture. An IaaS provider provides the storage space. Customers store their personal as well as business data on remote cloud. But in the absence of security of data, an unauthorized user can access the data resulting great loss of customer.

Generally, there are following three requirements of data security:

Data Confidentiality: Confidentiality is primary and one of the major aspects of data security. Confidentiality of data refers to only authorized users are allowed to access the data. Data confidentiality ensures the customers that their data on cloud will not be accessed by any unauthorized user.

Data Integrity: Data Integrity refers to protecting data from unauthorized modification. Data integrity must be implemented on cloud so that data cannot be altered illegitimately.

Data Availability: Availability is another critical aspect of data security. The goal of data availability is to ensure customers that they can access their data anytime from anywhere using network safely.

Many solutions to the problem of data security in cloud computing which implements standard traditional encryption methods are proposed in [5][6]. Authors of these papers suggested storing encrypted data on cloud data server and also exchanging encrypted data between customers' local machine and cloud data server. Another solution which is based on third party is also proposed in [9]. But these solutions are limited to only when data is in storage state and in transmission state i.e. encrypted data can be stored on cloud and exchanged between cloud and customers' machine. But what if when data is in processing state, if a user needs to perform operations on data? Operations cannot be performed on encrypted data directly. Therefore, data encrypted using standard encryption methods require decryption to be operated. Hence, decryption of data and performing operation on cloud make it available to cloud provider violating confidentiality of data.

Another hypothesis is that customer will operate data at local machine i.e. download encrypted data at local machine from cloud server, decrypt it, perform operations then again encrypt resultant data and finally store at cloud data server. But performing repeatively encryption/decryption make such solution very costly and practically infeasible. Another problem with this hypothesis is continuously exchanging of data over network makes it vulnerable to capture.

Another way to secure data is that to perform operations on encrypted data on cloud server. Developing such a solution introduces new methods of encryption which is called *Homomorphic Encryption.* Homomorphic encryption technique is a kind of encryption that allows operating encrypted data, assuring confidentiality of data during processing of data.

III. HOMOMORPHIC ENCRYPTION

A homomorphic encryption technique allows user to operate ciphertext directly. When user decrypts the resultant cipher, it is same as if operations are carried out on plaintext. Thus, making use of homomorphic encryption assures customers that their data is secure in all state: storage, transmission and processing.

Basic concept of homomorphic encryption is shown in Figure-1. Suppose a user want to add two numbers 10 and 15, which are stored on cloud data server in encrypted form (assume 10 as 100 and 15 as 150 respectively). The cloud server add two numbers and store sum as 250, that user download from the cloud server and decrypt it to the final answer 25.

Standard symmetric encryption methods such as DES, AES do not follow the principle of homomorphism.

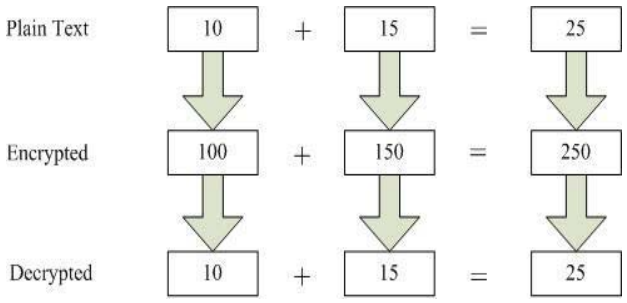

Figure-1: Homomorphic Encryption

Homomorphic encryption can either be fully or partial homomorphic encryption. A fully homomorphic encryption (FHE) supports arbitrary number of both operations addition as well as multiplication on ciphertext. For example, user can perform an encrypted search on Internet: encrypt input, send it to search engine that performs multiple additions and multiplications on ciphertext and returns decrypted searched results. Craig Gentry gave first FHE in 2009[10]. Homomorphic encryption schemes proposed before Craig Gentry are partial homomorphic encryption schemes.

On the other hand, partial homomorphic encryption

scheme support either addition operation (known as additive homomorphic encryption scheme) or multiplication operation (known as multiplicative homomorphic encryption scheme). Such scheme can perform limited number of both operations. For example a partial homomorphic encryption schemes can support multiple addition with single multiplication.

Paillier cryptosystem [12] scheme is a partial homomorphic encryption schemes that support only addition operation. Similarly, RSA cryptosystem [13] scheme support only multiplication. Boneh-Goh-Nissim cryptosystem [16] is additive and multiplicative homomorphic encryption that supports unlimited number of additions but single multiplication.

A. Paillier Cryptosystem

Paillier cryptosystem was given by Pascal Paillier in 1999. Paillier cryptosystem is a public key cryptosystem. Here, we are not describing cryptographic notations used in algorithm. Readers can refer to [12].

1) Algorithm:

a) Key Generation:

(i) Select two larger prime number p and q
(ii) Compute n = p*q and λ = lcm(p-1, q-1)
(iii) Select $g \in Z^{*}_{n^2}$ | gcd ($L(g^{\lambda} \bmod n^2)$,n) = 1, where L(u) = (u-1)/n
(iv) Compute $\mu = L(g^{\lambda} \bmod n^2)^{-1} \bmod n$
(v) Select $P_k(n, g)$ as public key and $S_k(\lambda, \mu)$ as private key

b) Encryption:

(i) Message 'M' is integer number $M \in Z_n$
(ii) Choose a random $r \in Z_n^{*}$
(iii) Encrypt 'M' using public key $P_k(n, q)$, $C = g^M.r^n \bmod n^2$

c) Decryption:

(i) Decrypt 'C' using private key $S_k(\lambda, \mu)$, $M = L(C^{\lambda} \bmod n^2). \mu \bmod n$

2) Additive Homomorphic property of Paillier Cryptosystem:
An encryption scheme is said to be additive homomorphism encryption scheme if

$$E(x+y) = E(x).E(y)$$

Suppose C_1 and C_2 are two ciphers

$$C_1 = g^{M1}.r_1^{n} \bmod n^2$$

$$C_2 = g^{M2}.r_2^{n} \bmod n^2$$

then

$$C_1. C_2 = g^{(M1+M2)}.(r_1.r_2)^{n} \bmod n^2$$

Hence, Paillier cryptosystem realizes property of additive homomorphism encryption.

B. RSA Cryptosystem

RSA system is given by R. Rivest, A. Shamir, and L. Adleman. RSA is a public key cryptosystem.

1) Algorithm:

a) Key Generation:

(i) Select two large prime number p and q
(ii) Compute n = p*q and $\Phi(n) = (p-1).(q-1)$
(iii) Select e | gcd(e, $\Phi(n)$) = 1
(iv) Compute d | (d,e) = 1 mod $\Phi(n)$
(v) Select $P_k(e, n)$ as public key and $S_k(d, n)$ as private key

b) Encryption:

(i) Message 'M' is integer number $M \in Z_n$
(ii) Encrypt 'M' using public key $P_k(e, n)$, $C = M^e \bmod n$

c) Decryption:

(i) Decrypt 'C' using private key $S_k(d, n)$, $M = C^d \bmod n$

2) Multiplicative Homomorphic property of RSA Cryptosystem:

An encryption scheme is said to be multiplicative homomorphism encryption scheme if

$$E(x.y) = E(x).E(y)$$

Suppose C_1 and C_2 are two ciphers

$$C_1 = M_1^{e} \bmod n$$

$$C_2 = M_2^{e} \bmod n$$

then

$$C_1. C_2 = (M_1.M_2)^{e} \bmod n$$

Hence, RSA cryptosystem realizes property of multiplicative homomorphism encryption.

C. Boneh-Goh-Nissim Cryptosystem

Boneh-Goh-Nissim cryptosystem [16] is a public key cryptosystem that was proposed by D. Boneh, E.-J. Goh, and K. Nissim in 2005. Boneh-Goh-Nissim is near to fully homomorphic encryption. It supports unlimited addition operation but one multiplication operation.

1) Algorithm:

a) Key Generation:

(i) Two prime numbers $q_1, q_2 \in Z$, $n = q_1 * q_2$
(ii) Two generator g, $u \in G$ and $h = u^{q2}$
(iii) Select $P_k(n,g,h,e,G,G_1)$ as public key and $S_k(q_1)$ as private key.(G, G_1: multiplicative group of order n and e: $G \times G1 \rightarrow G1$ is bilinear map)

b) Encryption:

(i) Encrypt 'm' using public key P_k, $C=g^m.h^r \bmod n$

c) Decryption:

(i) To decrypt 'C' using private key $S_k(q_1)$, perform

$C^{q1} = (g^{q1})^m$, i.e. message m is discrete logarithm of C^{q1} to the base g^{q1}.

2) Additive Homomorphic property of Boneh-Goh-Nissim Cryptosystem:

Suppose C_1 and C_2 are two ciphers

$$C_1 = g^{M1}.h^{r1} \bmod n$$

$$C_2 = g^{M2}.h^{r2} \bmod n$$

then

$$C_1. C_2 = g^{(M1+M2)}.h^{(r1+r2)} \bmod n$$

Hence, Boneh-Goh-Nissim cryptosystem realizes property of additive homomorphism encryption.

IV. HOMOMORPHIC ENCRPTION IN CLOUD

Homomorphic Encryption (HE) is the new concept that allow user to operate encrypted data. Therefore, homomorphic encryption technique seems as solution to the problem of data security in processing state.

Partial Homomorphic encryption scheme such as RSA cryptosystem and Paillier cryptosystem are insufficient for cloud computing. Partial HE is limited to number of operations. Partial HE can be either additive HE or multiplicative HE. RSA cryptosystem is vulnerable to common modulus attack. Suppose C_1 and C_2 are two ciphers encrypted with two RSA public keys (e, n) and (f, n) {gcd (e, f) =1} respectively. If attacker is able to capture C_1 and C_2, he /she can extract M_1 and M_2.

Craig Gentry created first FHE scheme in 2009. But it was unable to run on any hardware till 2010. On the other hand, Craig Gentry also showed that it takes 36 hours to AES evaluate in 2012. However, this time is reduced by Homomorphic Encryption library (Helib) [14]. But it did not achieve a feasible time to perform operation on data over Internet. Helib is a software library, developed by IBM in 2013 that implements fully homomorphic encryption technique. Currently available Helib is an implementation of the Brakerski-Gentry-Vaikuntanathan scheme [15] along with optimizations to run faster.

V. CONCLUSION

In this paper, we discussed data security in processing state is one of the major barrier in the adaptation of cloud computing. We also discussed three partial homomorphic encryption methods and their applications in cloud computing. Traditional encryption techniques are not sufficient to secure data in processing state. It appears that homomorphic encryption methods detract the problem of data security in cloud computing. But currently both fully as well as partial homomorphic methods are not feasible and not so easy to implement for cloud computing. Authors of [11] showed that fully homomorphic method took long time to operate on encrypted data.

On the other hand, partial homomorphic methods are also not efficient as it is able to perform either addition or multiplication.

At last in present, homomorphic encryption based solutions for data security are not practical in real world. Therefore, the problem will lead to design efficient and feasible homomorphic encryption algorithms.

REFERENCES

[1]. D. Zissis and D. Lekkas, "Addressing Cloud Computing Security Issues", in Future Generation Computer Systems, Elsevier journal , vol. 28, pp: 583-592, 2012.

[2]. M. Zhou, R. Zhang, W. Xie, W. Qian and A. Zhou, "Security and Privacy in Cloud Computing: A Survey", in proceeding of IEEE International Conference on Semantics, Knowledge, and Grids, Beijing, China, pp: 105-112, 2010.

[3]. S. Subashini and V. Kavitha, "A Survey on Security issues in Service delivery models of Cloud Computin", Journal of Network and Computer Applications, Elsevier, vol. 34, pp: 1-11, 2011.

[4]. H. Tianfield, "Security Issues In Cloud Computing", in proceeding of IEEE International Conference on Systems, Man, and Cybernetics, Seoul, Korea, pp: 1082-1088, 2012.

[5]. C. Yang, W. Lin and M. Liu, "A Novel Triple Encryption Scheme for Hadoop-based Cloud Data Security", in proceeding of IEEE International Conference on Emerging Intelligent Data and Web Technologies, Xi'an, China, pp: 437-442, 2013.

[6]. P. Rewargad and Y. Pawar, "Use of Digital Signature with Diffie Hellman Key Exchange and AES Encryption algorithm to Enhance Data Security in Cloud Computing", in proceeding of IEEE International Conference on Communication Systems and Network Technologies, Gwalior, India, pp: 437-439, 2013.

[7]. D. Hrestak and S. Picek, "Homomorphic Encryption in the Cloud", in proceeding of IEEE International Convention on Information and Communication Technology, Electronics and Microelectronics, Opatija, Croatia, pp:1400-1404, 2014.

[8]. M. Tebaa, Said E. Hajji and A. E. Ghazi, "Homomorphic Encryption method applied to Cloud Computing", in proceeding of IEEE National Days of Network Security and Systems, ENSA/Marrakech, Morocco, pp: 86-89, 2012.

[9]. S. Han and J. Xing, "Ensuring Data Storage Security Through A Novel Third Party Auditor Scheme in Cloud Computing", in proceeding of IEEE International Conference on Cloud Computing and Intelligent Systems, Beijing, China, pp: 264-268, 2011.

[10]. Craig Gentry, A Fully Homomorphic Encryption Scheme, Sep-2009, available at http://crypto.stanford.edu/craig/craig-thesis.pdf.

[11]. C. Gentry and S. Halevi, N.P. Smart, "Homomorphic Evaluation of the AES Circuit", in proceeding of CRYPTO, Santa Barbara, California, USA, pp:850-867, 2012.

[12]. P. Paillier, "Public-key Cryptosystems based on Composite Degree Residuosity Classes", in proceedings of the International Conference on Theory and Application of Cryptographic Techniques, EUROCRYPT, Berlin, Heidelberg: Springer-Verlag,, pp. 223–238, 1999.

[13]. R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-key Cryptosystems", Communications of the ACM, vol. 21, pp: 120–126, 1978.

[14]. S. Halevi and V. Shoup, "Design and Implementation of a Homomorphic-Encryption Library", April-2013, available at http://people.csail.mit.edu/shaih/pubs/he-library.pdf.

[15]. Z. Brakerski and V. Vaikuntanathan, "Efficient Fully Homomorphic Encryption from (Standard) Learning with Error (LWE)", in proceeding of IEEE Annual Symposium on Foundation of Computer Science, Palm Spring, California, USA, pp: 97–106, 2011.

[16]. D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-dnf formulas on Ciphertexts", in Proceedings of the Second International Conference on Theory of Cryptography, ser. TCC'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 325–341.