International Journal of Computer Sciences and Engineering Open Access
Survey Paper Volume-04, Issue-01 E-ISSN: 2347-2693

Survey on Securing Data using Homomorphic Encryption in Cloud Computing

Suraj S. Gaikwad[1*], Amar R. Buchade[2]

[1*,2]*Department of Computer Engineering,*
PICT, Savitribai Phule Pune University, India

www.ijcseonline.org

Received: Dec/12/2015 Revised: Dec/24/2015 Accepted: Jan/18/2016 Published: 30/Jan/2016

Abstract- Cloud computing is technique which has become today's hottest research area due to its ability to reduce the costs associated with computing. In today's Internet world, it is most interesting and enticing technology which is offering the services to various users on demand over network from different location and using range of devices. Since Cloud Computing stores the data and propagate resources in the open environment, security has become the main obstacle which is hampering the deployment of Cloud environments. So To ensure the security of data, we proposed a method which uses Multilayer Security for securing Data using Homomorphic Encryption in Cloud Computing. Homomorphic encryption is a form of encryption that perform computations on ciphertext thus generating an encrypted result which, when decrypted, matches the result of operations performed on the plaintext. This System helps to secure Data for various cloud users.

Keywords- Cloud Computing, Privacy Preservation, Homomorphic Encryption

I. INTRODUCTION

Need of cloud to manipulate and manage data is increasing rapidly for sharing resources [4]. It is financially beneficial to store data with a third party, the cloud provider. However, storing data on third party infrastructure poses risks of data disclosure during retrieval. Therefore, the data is stored in encrypted form [9]. Encryption alone is not sufficient, as it provides security but reduces usability. Major advantage to be drawn from cloud computing is due to delegation of computation, but encrypting data would require sharing of keys with the third party performing computation on it, thereby increasing vulnerability. Hence, there is a need of feasible homomorphic schemes that allow user to compute on encrypted data, to verify a computation done by third party, to search an encrypted database, and so on[16].Fully Homomorphic Encryption allows a third party to evaluate arbitrary functions over encrypted data without decrypting it[1, 2].

Cloud provides large shared resources where users can enjoy the facility of storing data or executing applications. Instead of gaining benefit of large resources, storing critical data in cloud is not secured [3, 4]. Hence, cloud security is one of the important topics to make cloud useful at the enterprise level. Data encryption is a primary solution for providing discreetness to sensitive data. However, processing of encrypted data requires extra overhead, since repeated encryption-decryption need to be performed for every simple processing on encrypted data. Hence, direct processing on encrypted cloud data is advantageous, which is possible due to the use of technique called Homomorphic encryption schemes [4, 5]. Homomorphic Encryption provides methods to perform operations directly on encrypted data.

A. Data Storage

One of the major concern while storing data at third party data center is that user is unaware about the location of data where exactly it is stored [13].Users rely on services that are non-transparent to them, and no information about the servers operation is broadcast. Although this can improve security by insignificance, it also undermines user trust [13, 15]. How the security of data is ensuring at the server side might not be clear to clients. Data retention is also a concern for users. The cloud service provider can be able to keep deleted data in backups or for some unpublished reason. For example, Facebook is able to kept deleted data but removed it from view. The same concept also applies when a service is terminated. So while storing data at third party data center not only securing data from unauthorized user but securing it from cloud service provider is also important. So while performing operation data may not be accessible to service provider is also important.

B. Access Control

Most of the cloud systems include basic access control [10, 11]. Almost every system has privileged users, such as system administrators who is able to access user data and all application over the machine. When data or processes are outsourced to the third party data centers, possibly sensitive

data or processes are handed over for safe keeping [13]. In a local setting, user know to whom they trust for their data, but in a cloud setting users rarely know the location of the cloud server, the people managing it at the server side, and who has access to it in general.

Insider threats are particularly concerning, because such attacks can lead to enormous damage. Malicious employees can cause major harm, but even carelessness can inflict damage by allowing outside attackers to obtain insider privileges. Cloud services are interesting targets for criminals, because a successful attack can yield a large amount of information. Attacks can vary from inappropriate access of information to reveal or altering personal data [13]. A privacy leak can be damaging by itself, but publishing or forging personal information can cause even more serious harm.

So it is important for user to know who is accessing his/her data and for what purpose and violation of data by cloud service provider should not be happen.

C. *Identity Protection*

Data traveling over the Internet provides valuable information about people. Search keywords, credit card usage, and mobility patterns are just a few examples of information that can be used to identify and track individuals from supposedly anonymzed data, attackers are able obtain this information [13,10].This same data is available practically without restrictions to cloud service providers. For example, some cloud service providers business models include targeted advertising that is based on monitoring account traffic or data stored on user accounts [13].

D. *Privacy Preservation*

One of the problems in public clouds is how to share documents based on fine-grained attribute-based access control policies (acps) [7, 10]. Solution is to encrypt documents which satisfy different policies with different keys using a public key cryptosystem such as attribute-based encryption, and/or proxy re-encryption. But there are some weaknesses in these approaches, it cannot efficiently handle adding/revoking users or identity attributes, and policy changes, It needs to keep copies of same documents, which causes to have high computational costs [11, 13]. A direct application of a symmetric key cryptosystem, where users are grouped together based on the policies they satisfy and unique keys are assigned to each group, also has similar weaknesses. In order to maintain the user data privacy homomorphic encryption is best approach where all operation are carried out on encrypted data without dataloss or unauthorized data access.

II. SECURITY ISSUES IN CLOUD

To provide security to the data is always having a importance issue because of the critical nature of the cloud and very large amount of complicated data it carries, the need is even important [12]. Therefore, data security and privacy issues that need to be solved have they are acting as a major obstacle in adopting cloud computing services.

The main security issues of cloud are:

A. *Privacy and confidentiality*

Once user outsources data over the cloud there should be having some guarantee that data is accessible only to the authorized user and unauthorized access are prevented as well as data is protected from cloud service provider also [10, 12]. The cloud user should be confident that data stored on the cloud will be confidential.

B. *Security and data integrity*

Data security can be provided using different encryption and decryption techniques. With providing security to data, cloud service provider should also implement technique to monitor integrity of data on the cloud [12, 13].

III. TOWORDS HOMOMORPHIC ENCRYPTION

A. *Basics about encryption*

In this section, we will recall some important concepts concerning encryption schemes. Encryption schemes are, first and foremost, designed to preserve confidentiality. In cryptography, encryption is the process of encoding messages or information in such a way that only authorized parties can read it [6]. Encryption does not of itself prevent interception, but denies the message content to the interceptor. In an encryption scheme, the intended communication information or message, referred to as plaintext, is encrypted using an encryption algorithm, generating ciphertext that can only be read if decrypted [6].

B. *Symmetric encryption schemes*

In symmetric crypto system encryption and decryption are performed with the same key as shown in figure1. Hence, the sender and the receiver have to use same key before performing any secure communication. Therefore, it is not possible for two different people who never met each other to use such schemes directly [6]. This also implies to share a different key with every one we want to communicate with. Nevertheless, symmetric schemes present the advantage of being really fast and are used as often as possible.

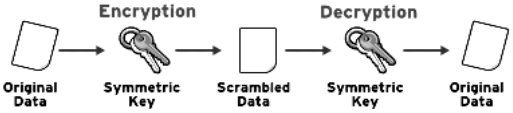

Fig.1. Symmetric Encryption Scheme [18]

C. Asymmetric encryption schemes

In relation with previous scheme, asymmetric schemes introduce a fundamental difference between the abilities to encrypt and to decrypt as shown in figure 2. The encryption key is public, as the decryption key remains private. When Bob wants to send an encrypted message to Alice, she uses her public key to encrypt the message. Alice will then use her private key to decrypt it. Such schemes are more functional than symmetric ones since there is no need for the sender and the receiver to agree on anything before the transaction [6]. These schemes, however, have a big drawback: they are based on nontrivial mathematical computations, and much slower than the symmetric ones.

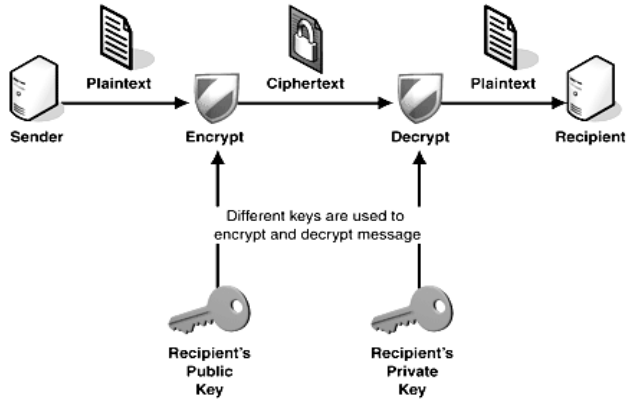

Fig. 2. Asymmetric Encryption Scheme [19].

Trends in Solutions

A number of solutions to the problem of data security and privacy in the cloud have been proposed.

D. Securing the data with Homomorphic Encryption

While using cloud storage a major concern of cloud users is the potential for losing data privacy once the data has moved to the cloud [13, 15]. Customers need guarantee that their data is well protected by cloud service providers. Encryption can reduce this fear, but it also has drawbacks. To avoid time-consuming downloading and uploading of data for customers, the cloud provider can be able to perform operations in the cloud [13, 14]. However, to manipulate encrypted data in the cloud, users must need to share their encryption/decryption keys with the cloud provider, which allow effective access to their data.

One of the top threats to cloud computing is malicious insiders [13, 15]. An insider can be a rogue administrator employed by a cloud service provider, an employee of the victim organization who exploits vulnerabilities to gain unauthorized access, or an attacker who can make use of cloud resources to perform attacks. The multitenant nature of the cloud computing environment makes it difficult to detect and prevent insider attacks.

Homomorphic encryption is technique to perform operation on encrypted data which is also known as ciphertext, which can generate an encrypted result, which, when decrypted, matches the result of the same operations performed on the original data also known as plaintext [14, 15, 16]. What exactly it does is shown in following figure 3 [16]. This can be a major advantage for applications that outsource encrypted data to the cloud.

Homomorphic encryption technique is striking for many applications, but it has some serious limitation: the homomorphic property is typically restricted to only one operation, which is usually addition or multiplication [14]. Methods having the homomorphic property for both addition and multiplication operation simultaneously bring us a step closer to real-life applications.

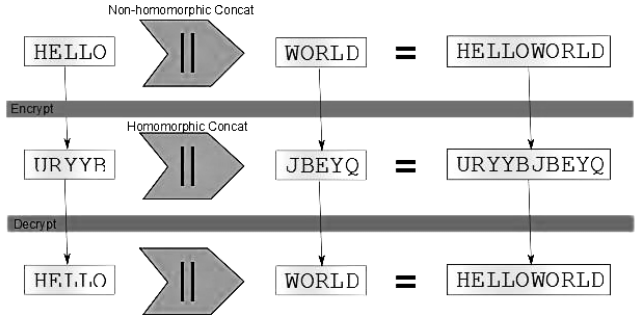

Fig. 3. A string concat example of Homomorphic Encryption [16].

Ronald Rivest and his colleagues introduced the concept of fully homomorphic encryption under the name privacy homomorphism in 1978, but it was only in 2009 that Craig Gentry proposed a fully homomorphic encryption (FHE) scheme. Gentry's scheme allowed an arbitrary number of additions and multiplications on encrypted data while guaranteeing that the results were correctly reflected in the decrypted data.

IV. FUNCTIONS OF HOMOMORPHIC ENCRYPTION

Homomorphic Encryption H is a set of four functions as shown in figure 4 [14].

H= { *KeyGeneration, Encryption, Decryption, Evaluation}*

- KeyGeneration [14]: client will generate pair of keys public key P_k and secret key S_k for encryption of plaintext.
- Encryption [14]: Using secret key S_k client encrypt the plain text P_t and generate $ES_k(P_t)$ and along with public key P_k this cipher text C_t will be sent to the server.

$$C_t = E(S_k(P_t))$$

- Evaluation [14]: Server has a function *f* for doing evaluation of cipher text C_t and performed this as per the required function using P_t.

$$Eval_{Pk}(f, ES_k(a), ES_k(b))$$
$$Eval_{Pk}(f, C_{t1}, C_{t2})$$

- Decryption [14]: Generated $Eval(f(P_t))$ will be decrypted by client using its Sk and it gets the original result.

$$Result = DS_k(Eval_{Pk}(f, ES_k(a), ES_k(b)))$$
$$\text{i.e., } Result = DS_k(Eval_{Pk}(f, C_{t1}, C_{t2}))$$
$$PT = DecS_k(P_k, C_t)$$

Following figure 4 [14] explain the detain flow of homomorphic encryption function.

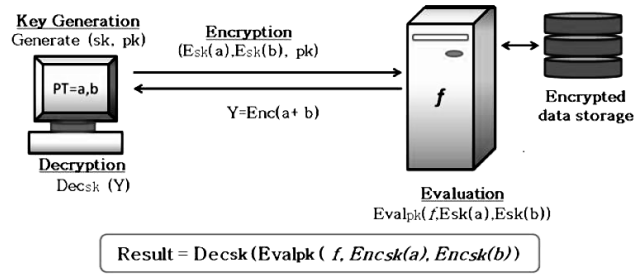

Fig. 4. Homomorphic Encryption functions [14].

V. PROPERTIES OF HOMOMORPHIC ENCRYPTION

An encryption is homomorphic, if [14, 15]: from E_{nc} (a) and E_{nc} (b) it is possible to compute E_{nc} (*f* (a, b)), Where *f* can be: +, x and without using the private key. Among 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 Pailler and Goldwasser-Micalli cryptosystems, and the multiplicative Homomorphic encryption (only products on raw data) is the RSA and El Gamal cryptosystems.

- Additive Homomorphic Encryption: A Homomorphic encryption is additive, if [14]:

$$E_k(P_{t1} + P_{t2}) = E_k(P_{t1}) + E_k(P_{t2})$$

Where P_{t1} and P_{t2} are plaintext messages,
+ denote the additive homomorphic operation.

- Multiplicative Homomorphic Encryption: Homomorphic encryption is multiplicative, if:

$$E_k(P_{t1} \times P_{t2}) = E_k(P_{t1}) \times E_k(P_{t2})$$

Where P_{t1} and P_{t2} are plaintext messages,
X denotes the multiplicative homomorphic operation.
The encryption function is semantically secure which means that an adversary cannot get any additional information about the plaintext from a given set of ciphertexts and corresponding public key [15].

VI. PROPOSED SYSTEM

Confidentiality of information in a public cloud is a major concern, which is guaranteed by suitable encryption algorithm. However, for every single processing on encrypted data, information should be downloaded and decrypted in client side and after processing it is further encrypted and uploaded to cloud. These obvious needs for repeated decryption encryption increase the processing complexity and out-weigh the advantage of using cloud resources. Since the computation is done in the client end, the objective of utilizing large processing power at the cloud-end is defeated and the ciphertext is continuously exposed to the adversary. Hence one needs to develop a mechanism in which arbitrary algorithms can be executed over encrypted data directly on the cloud side. This can be achieved using the technique called Homomorphic Encryption. Hence our goal is develop a system for securing data in cloud computing environment using Homomorphic Encryption approach that is to be done at various levels i.e., at user level and at CSP level.

CONCLUSION

The cloud computing security based on Homomorphic encryption, which is a new concept of security which enables providing results of calculations on encrypted data without knowing the raw data on which the calculation was carried out, with respect of the data confidentiality. In this paper we show a state of the art on homomorphic encryption schemes discuss its parameters, Its additive, multiplicative property and performances also discuss the data security issues in cloud computing. The application of Homomorphic encryption is an important stone in Cloud Computing security, more generally, we the user outsource the calculations on confidential data to the Cloud server, keeping the secret key that can decrypt the result of calculation. So, security problem is overcome through this Homomorphic algorithm. Data confidentiality is managed by this algorithm.

REFERENCES

[1] Craig Gentry,"A FULLY HOMOMORPHIC ENCRYPTION SCHEME ," September **2009**.
[2] Craig Gentry, Shai Halevi, " Implementing Gentry's Fully-Homomorphic Encryption Scheme," IBM Research, February 4, **2011**.
[3] Craig Gentry, "Computing Arbitrary Functions of Encrypted Data," IBM T.J. Watson Research Center, ACM **2008**.
[4] Ayantika Chatterjee, Indranil Sengupta, "Translating Algorithms to handle Fully Homomorphic Encrypted Data on the Cloud," IEEE Transactions on Cloud Computing, **2015**.
[5] Nitesh Aggarwal, Dr CP Gupta, "Fully Homomorphic Symmetric Scheme Without Bootstrapping," International

Conference on Cloud Computing and Internet of Things, **2014**.
[6] Murat Kantarcioglu, Wei Jiang, "A Cryptographic Approach to Securely Share and Query Genomic Sequences," IEEE Transactions,VOL. 12, NO. 5, Sep **2008**.
[7] David W. Archer, Kurt Rohloff, "Computing with Data Privacy:Steps toward Realization," IEEE Computer and Reliability Societies jan/feb **2015**.
[8] FENG Chao, XIN Yang, "Fast key generation for Gentry-style homomorphic encryption," ScienceDirect, December **2014**.
[9] Monique Ogburn, Claude Turner, "Homomorphic Encryption," ScienceDirect, **2013**.
[10] Mohamed Nabeel, Elisa Bertino, "Privacy Preserving Delegated Access Control in Public Clouds," IEEE Transactions, Volume: 26, april **2013**.
[11] Mohamed Nabeel, Ning Shang, "Privacy Preserving Policy-Based Content Sharing in Public Clouds," IEEE Transactions, Volume. 25, NO. 11, November **2013**.
[12] Jiadi Yu, Peng Lu, "Toward Secure Multikeyword Top-kk Retrieval over Encrypted Cloud Data," IEEE Transactions, Volume. 10, NO. 4, July/August **2013**.
[13] Zahir Tari, Xun Yi, Uthpala S. Premarathne, "Security and Privacy in Cloud Computing: Vision, Trends, and Challenges," IEEE Cloud Computing, **2015**.
[14] Payal V. Parmar, Shraddha B. Padhar, "Survey of Various Homomorphic Encryption algorithms and Schemes," International Journal of Computer Applications,Volume 91. No.8, April **2014**.
[15] Bharath K. Samanthula, Yousef Elmehdwi, "A secure data sharing and query processing framework via federation of cloud computing," Information Systems, ScienceDirect, **2013**.
[16] Shashank Bajpai, Padmija Srivastava, "A Fully Homomorphic Encryption Implementation on Cloud Computing," IJICT, Volume 4, No.8, 2014.
[17] Jian Li, Danjie Song, Sicong Chen, "A simple fully Homomorphic encryption scheme available in cloud computing," IEEE 2nd International Conference Volume:01, Oct **2012**.
[18] https://mdn.mozillademos.org/files/10303/05scrypt2.png
[19] https://blog.malwarebytes.org/wpontent/uploads/2013/10/assemcrypto.gif

AUTHORS PROFILE

Suraj Gaikwad received his B.E degree from D.Y.Patil COE, Pune. He is ongoing M.E. student in PICT, Pune, India. His research interest includes Cloud Computing and Security.

Amar Buchade presently working as assistant professor in the Department of Computer Engineering at Pune Institute of Computer Technology, Pune. He received B.E. and M.E. in Computer Engineering from WCOE, Sangli, India. His research area includes Distributed System, Cloud Computing and Security.

