IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.301

Volume 3, Issue 2, July 2023

How Homomorphic Cryptosystem used for Cloud Database

Kedar S. Yele[1] and Mr. Prakash Sakharkar[2]
Student, Department of MCA[1]
Professor, Department of MCA[2]
Late Bhausaheb Hiray S.S. Trust's Institute of Computer Application, Mumbai, India
kedaryele0@gmal.com

Abstract: *The concept of cloud computing receiving a great deal of attention both in publication and among users. Cloud computing is the delivery of computing services over the Internet. Cloud services allow individuals and businesses to use software and hardware resources that are managed by cloud providers at remote locations. The distance between the client and the physical location of his data creates a barrier because this data can be accessed by a third party and this would affect the privacy of client's data. The using of traditional encryption schemes to encrypt the remote data before sending to the cloud provider has been most widely used technique to bridge this security gab. But, the client will need to provide the private key to the server to decrypt the data before perform the calculations required. Homomorphic encryption allows performing computations on encrypted data without decryption. This paper deals with the use of homomorphic encryption to encrypt the client's data in cloud server and also it enables to execute required computations on this encrypted data.*

Keywords: Homomorphic encryption

I. INTRODUCTION

Cloud computing is a hot topic in the information technology field. It enables users to get almost unlimited computing power and it offers potential benefits to these users in terms of instant availability, scalability and resource sharing. Examples of cloud services offers by cloud providers include online file storage (e.g. Drop box), social networking sites (e.g. Face book), webmail (e.g. Gmail), and online business application (e.g. Brokerage).The essential characteristics of cloud computing include on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service. On-demand self-service means that clients (users or organizations) can request and manage their own computing resources. Broad network access allows services to be offered over the Internet or private networks. Pooled resources mean that customers draw from a pool of computing resources, usually in remote data centers. Rapid elasticity means that services can be scaled larger or smaller. And use of a service is measured and customers are billed accordingly. Although cloud computing has become a mature service model, the adoption of its services by customers (businesses, consumers, etc.) is limited by concerns about the loss of privacy of their private data. Encryption of data could solve this issue, but if the clients want to manipulate their encrypted data in the cloud, they have to share the secret key with cloud provider to decrypt it before execute the required operations.Homomorphic encryption is theappropriate solution to solve cloud computing security issues, since its schemes enable to perform computations on encrypted data without sharing the secret key needed to decrypt the data.Homomorphic encryption technology supports the management of cipher text data under privacy protection. It can directly retrieve, calculate and count cipher text in the cloud and return the results to users in the form of cipher text. Compared with traditional encryption algorithms, homomorphism encryption technology does not require frequent encryption and decryption between the cloud and users, thus reducing the cost of communication and computing. Homomorphic encryption technology is the key technology to ensure the confidentiality of data in cloud environments. With the homomorphic features of homomorphic encryption technology, key security issues in cloud services can be solved, and the application of cloud computing can further promote the development of homomorphic encryption

DOI: 10.48175/568

technology. In this paper, single homomorphic encryption algorithms and fully homomorphic encryption algorithms were described and analyzed respectively. For the homomorphic algorithm, the common widely used Hill, RSA, Paillier, and ElGamal encryption algorithms were listed, which were analyzed from the aspects of algorithm description, homomorphism analysis, and performance. And the efficiency of encryption and decryption of these algorithms was compared. The research and application of these algorithms in the cloud environment were introduced. For the homomorphic encryption algorithm, several representative homomorphic algorithms and their specific application scenarios in the cloud environment were reviewed. The performance indicators were compared and analyzed, and the problems to be solved in the application of homomorphic encryption algorithms were pointed out.

II. SECURITY OF CLOUD COMPUTING

Cloud computing can significantly reduce the cost and complexity of owning and operating computers and networks. If an organization uses a cloud provider, it does not need to spend money on information technology infrastructure, or buy hardware or software licenses. Cloud services can often be customized and flexible to use, and providers can offer advanced services that an individual company might not have the money or expertise to develop.

2.1 Definition of Cloud Computing

The following definition of cloud computing has been provided by National Institute of Standards and Technology (NIST) of U.S. “Cloud computing is a model for enabling 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 or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models”.

2.2 Types of Deployment Models

- The Public Cloud: A public cloud can be accessed by any subscriber with an internet connection and access to the cloud space.
- The Private Cloud: A private cloud is established for a specific group or organization and limits access to just that group.
- The Community Cloud: A community cloud is shared among two or more organizations that have similar cloud requirements (e.g. security).
- The Hybrid Cloud: A hybrid cloud is a combination of at least two clouds, where the clouds included are a mixture of public, private, or community.

2.3 Types of Service Models

There are three service models of cloud computing as follows:

- Software as a Service (SaaS): A SaaS provider gives subscribers access to both resources and applications. SaaS makes it unnecessary for the user to have physical copy of software to install on his devices. SaaS also makes it easier to have the same software on all of user’s devices at once by accessing it on the cloud (e.g. web-based email),). In a SaaS agreement, users have the least control over the cloud.
- Platform as a Service (PaaS): A PaaS provider gives subscribers access to the components that they require to develop and operate applications over the internet. Like using the programming languages, libraries, services, and tools supported by the provider.
- Infrastructure as a Service (IaaS): An IaaS agreement deals primarily with computational infrastructure. In an IaaS agreement, the subscriber completely outsources the storage and resources, such as hardware and software (e.g. host firewalls).), that they need.

III. HOMOMORPHIC ENCRYPTION

Homomorphic encryption is a type of encryption that allows particular computations to be conducted on cipher text and return an encrypted result, the decrypted of result is equal the result of conducting the operation on the plaintext. The property of homomorphic is useful to develop a secure e- voting system with high privacy data retrieving scheme, also it makes the use of cloud computing by ensuring the privacy of processed data. An example for its mathematical consistency, if there are two numbers 10 and 20 then both are encrypted to 56 and 69 respectively, the addition operator gives a number with value 125, the decrypted of this value is 30.

DOI: 10.48175/568

3.1 History of Homomorphic Encryption

The concept of homomorphic encryption was suggested in 1978 by Ronald Rivest and Leonard Adleman. But for 30 years the progress is very slow. In 1982, Shafi Gold wasserand Silvio Micali proposed their encryption system that able to encrypt one bit in additive homomorphic encryption. Pascal Paillier 1999 suggested another additive homomorphic encryption. In 2005, Dan Boneh, Eu-Jin Goh and Kobi invented a security system of encryption which conducts only single multiplication but large number of additions. In 2009 , Craig Gentry construct a fully homomorphic encryption based system that able to conduct both of addition and multiplication in the same time.

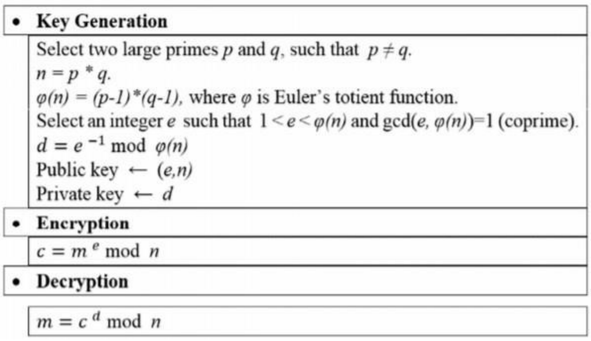
3.2 Categories of Homomorphic Encryption

There are two main categories of homomorphic encryption schemes: Partially Homomorphic Encryption (PHE) and Fully Homomorphic Encryption (FHE) schemes. PHE schemes, such as RSA, ElGamal, Paillier, Etc., allow performing either addition or multiplication on encrypted data. Construction of scheme supporting both operations simultaneously was elusive. Although Boneh et al. came closest, allowing unlimited additions and a single multiplication, It was not until2009 that the three decade old problem was solved in seminal work by Gentry , where he showed that performing both addition and multiplication simultaneously are possible in fully homomorphic encryption.

Partially Homomorphic Encryption A.

Multiplicative Homomorphic Schemes

A Homomorphic Encryption is multiplicative, if there is an algorithm that can calculate Enc(x × y) from Enc (x) and Enc (y) without knowing x and y . Such as RSA and ElGamal Algorithms. Figure 2 illustrates the RSA algorithm as an example of multiplicative.



- **Key Generation**

Select two large primes p and q, such that $p \neq q$.
$n = p * q$.
$\varphi(n) = (p\text{-}1)*(q\text{-}1)$, where φ is Euler's totient function.
Select an integer e such that $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n))=1$ (coprime).
$d = e^{-1} \bmod \varphi(n)$
Public key $\leftarrow (e,n)$
Private key $\leftarrow d$

- **Encryption**

$c = m^{e} \bmod n$

- **Decryption**

$m = c^{d} \bmod n$

Figure: RSA algorithm.

The multiplicative homomorphic property of RSA scheme

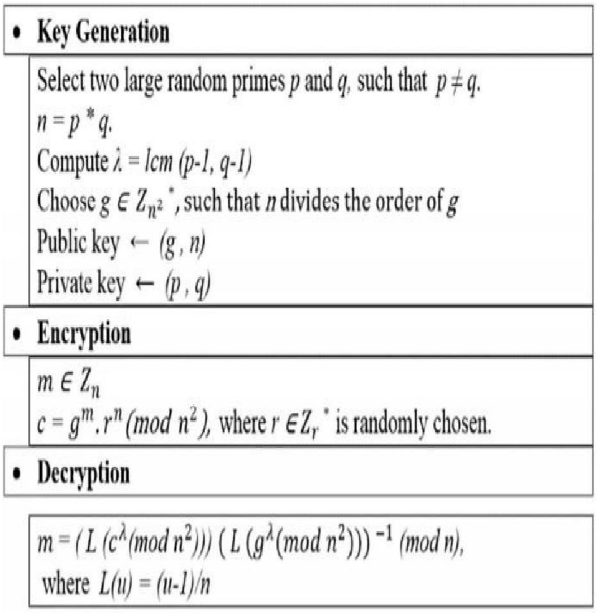
$$\text{Given } c_1 = m_1{}^{e} \bmod n,\ c_2 = m_2{}^{e} \bmod n$$

$$\begin{aligned} c_1 . c_2 &= E_{pk}(m_1). E_{pk}(m_2) \\ &= m_1{}^{e} . m_2{}^{e} \ (\bmod n) = (m_1 . m_2)^{e} \ (\bmod n) \\ &= E_{pk}(m_1 . m_2) \qquad (1) \end{aligned}$$

Additive Homomorphic Schemes

A Homomorphic Encryption is additive, if there is an algorithm that can calculate Enc(x + y) from Enc(x) and Enc(y) without knowing x and y .Such as Paillier and Goldwasser-Micali algorithms.

Figure 3 illustrate the Paillier algorithm as an example of additive homomorphic schemes.



- **Key Generation**

Select two large random primes p and q, such that $p \neq q$.
$n = p * q$.
Compute $\lambda = lcm\ (p\text{-}1, q\text{-}1)$
Choose $g \in Z_{n^2}{}^{*}$, such that n divides the order of g
Public key $\leftarrow (g, n)$
Private key $\leftarrow (p, q)$

- **Encryption**

$m \in Z_n$
$c = g^{m}.r^{n}(mod\ n^2)$, where $r \in Z_r{}^{*}$ is randomly chosen.

- **Decryption**

$m = (L\,(c^{\lambda}(mod\ n^2)))\ (L\,(g^{\lambda}(mod\ n^2)))^{-1}\ (mod\ n)$,
where $L(u) = (u\text{-}1)/n$

Fig 3. Paillier algorithm.

The homomorphic property of Paillier scheme can be

$$\begin{aligned} E(m_1). E(m_2) &= (g^{m_1} . r_1{}^{n})(g^{m_2} . r_2{}^{n}) \\ &= g^{m_1+m_2}(r_1 + r_2)^{n} \\ &= E(m_1 + m_2 (\bmod n)) \end{aligned}$$

Fully Homomorphic

Encryption

All of PHE schemes allow homomorphic computation of only one operation, either addition or multiplication, on encrypted date, except the Boneh-Goh-Nissim scheme which supports performing unlimited number of addition operation but only one multiplication. The constructing of a scheme that allows one to compute arbitrary computation (a scheme should allow an unlimited number of both addition and multiplication operations) over encrypted data has remained a central open problem in cryptography for more than 30 years and thought to be impossible until 2009, when Craig Gentry proposed the first plausible construction of a fully homomorphic scheme. Gentry's work is supporting multiplication and addition in the same time, correspond to AND (∧) and XOR (⊕) in Boolean algebra. The remarkable value of supporting these two Boolean functions is that any computation can be converted into a function that contains only (∧) and (⊕). In algebra, there are several techniques can be used to convert a function into more simple. By using this techniques can be convert a function to use only specific Boolean operation (e.g. ∧ or ⊕). For example ¬A can be expressed as A⊕ 1, another example is A ∨B, this can be converted into (¬A) ∧(¬B), then converted into (A ⊕ 1) ^ (B ⊕ 1). By utilizing such techniques, all functions can be converted into a series of (∧) and (⊕) operations. This is the basis of Gentry's work [19].Gentry is using lattice based cryptography. His proposed fully homomorphic encryption consists of several steps: start from what was called a somewhat homomorphic encryption scheme using ideal lattices that is limited to evaluating low- degree polynomials over encrypted data. It is limited because each cipher text is noisy in some sense, and this noise grows as one adds and multiplies cipher texts, until ultimately the noise makes the resulting cipher text indecipherable. Next, it squashes the decryption procedure so that it can be expressed as a low-degree polynomial which is supported by thescheme.Finally, it applies a bootstrapping transformation, through a recursive self-embedding, to obtain a fully homomorphic scheme.

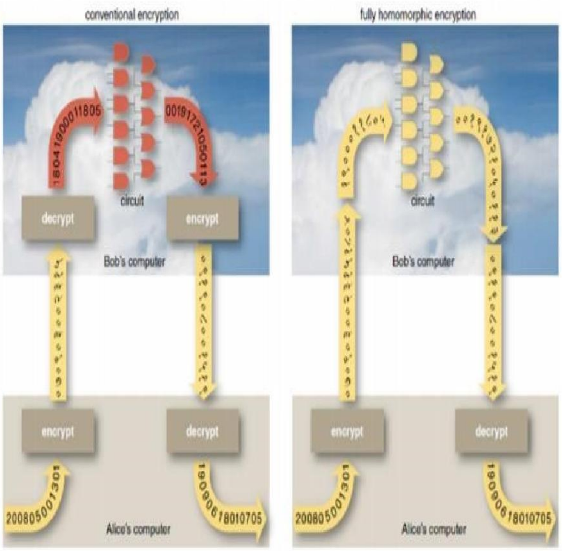


Figure 4 illustrates the difference between the conventional encryption schemes (not PHE) and fully homomorphic scheme.

IV. CONCLUSION

The security issues are a big problem for cloud computing development. To preserve the privacy of his data, the user must encrypt data before being sent to the cloud. Cloud computing security based on homomorphic encryption schemes, because these schemes allow performing computations on encrypted data without the need to the secret key. Partially Homomorphic Encryption (PHE) such as RSA and Paillier schemes are insufficient to secure cloud computing because these schemes allow to perform only one operation (either addition or multiplication) on the encrypted data of client. Fully Homomorphic Encryption is the best solution to secure the client data in cloud computing because its schemes enable to perform arbitrary computations on encrypted data without decrypting. DGHV and Gen10 schemes of FHE are insecure when they are used in cloud computing to secure data of client. SDC is a simple and considered efficient scheme to secure data in cloud computing. This paper analyzed some of the existing homomorphic encryption schemes and discussed the use of the most efficient one, SDC scheme, to secure cloud computing data. Future work will focus on implementation of SDC scheme in cloud computing and analysis the complexity of the scheme

REFERENCES

[1]. P. Mell, T. Grance, "The NIST Definition of Cloud Computing," National Institute of Standards and Technology, U. S. Department of Commerce, (2011).

[2]. J. Li, D. Song, S. Chen, X. Lu, "A Simple Fully Homomorphic Encryption Scheme Available in Cloud Computing", In Proceeding of IEEE, (2012).

[3]. M. van Dijk, C. Gentry, S. Halevi, V. Vaikuntanathan, "Fully homomorphic encryption over the Integers", in Proceedings of Advances in Cryptology, EUROCRYPT'10, pages 24–43, 2010