Journal of Theoretical and Applied Information Technology
20th December 2015. Vol.82. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SECURE DATA IN CLOUD COMPUTING USING HOMOMORPHIC ENCRYPTION

[1]YASMINA BENSITEL, [2]RAHAL ROMADI

[1]Ph.D student, RITM team, ENSIAS, Mohammed V- Souissi University, Rabat, Morocco

[2]Assistant Prof., RITM, ENSIAS, Mohammed V- Souissi University, Rabat, Morocco

E-mail: [1]yasminabensitel@gmail.com, [2]romadi@ensias.com

ABSTRACT

The emergence of cloud computing and cyber-physical systems made of security in processing data a major challenge. In order to ensure privacy and confidentiality of the data being manipulated, the use of cryptography is widely used today. In 2009, C. Gentry proposed the first fully homomorphic cryptosystem, to perform calculations on data previously encrypted without having to decrypt. This progress has allowed the opening of many industrial and research perspectives. However, despite recent progress, many limitations remain today on the lack of performance of these systems and their strong memory requirements. In this paper we focus on cloud computing along with its various security and privacy issues, we describe the role of homomorphic encryption scheme for ensuring data privacy and compare its types based on different characteristics.

Keywords: *Cloud computing, Homomorphic encryption, Data privacy, Confidentiality, Security*

1. INTRODUCTION

Cloud computing marks a new step towards IT infrastructure dematerialization; and gets a lot of attention, both in publications and among users. Whether they realize it or not, many people use cloud computing services for their own personal needs. For example, many people use social networking sites or webmail, and these are cloud services. Users of cloud computing are gaining autonomy, ergonomics and simplicity.

This new paradigm renders the Internet a large repository where resources are globally networked, easily shared and available to everyone as services. Virtualization is amongst the technologies used to provide these cloud services. Virtualization is a set of hardware and software techniques that allow to run multiple operating systems at the same time on one device completely separate from one another. Thus, an operating system called "host" is installed on a machine and hosts operating systems "guests" or "virtual machines". Virtualization and consolidation can simplify data-center management, reducing the number of machines by optimizing resource utilization and enabling high availability. Cloud security challenges are a problem for many researchers; first priority was to focus on security, which is the biggest concern of organizations considering a move to the cloud. But the adoption of the cloud applies only if security concerns are ensured. The question now is how can we guarantee privacy in cloud field?

The answer is the encryption, an encryption that is fully homomorphic, and allows to compute over encrypted data without having to decipher them. This type of encryption was proposed for the first time in 2009 at Stanford University by C. Gentry [6]: first cryptosystem providing the ability to perform arbitrary calculations on encrypted data without having to decipher them. Although the proposed solution has several drawbacks (very expensive in terms of memory and very slow in terms of speed), but has paved the way for numerous studies on this type of homomorphic encryption.

Our work is in line with this work, specifically around those of Sai Deep Tetali, who proposed MrCrypt [11]: a system that ensures confidentiality of data by executing processing clients on figures, this by using only partial homomorphic encryption algorithms.

This paper is organized as follow. In Section II, we give a brief description of cloud computing based on the definition given by NIST. Section III addresses threats and its security challenges. In Section IV, we provide background information on homomorphic encryption followed by an analysis of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

its performance in Section V. In Section VI we provide details of our implementation, and finally Section VII is devoted for the conclusion.

2. DEFINITION OF CLOUD COMPUTING ACCORDING TO NIST

Cloud computing is a technical environment that allows access to the application, via secure internet network to a shared set of computing resources. Hardware infrastructure (servers, network, storage, calculation capabilities, and availability), user applications (email, office automation, CRM, ERP) and services (security, interoperability) are thus shared in a virtualized computing platform network. All these means can be designed on demand, without material constraints, licenses or changes in production. Evolution is always guaranteed without interfering with the user's work processes.

The availability of these resources to the user results in a considerable gain in productivity. The result is broken down into elementary exclusively final solutions, flexible, ergonomic and intuitive.

Cloud computing is defined by NIST, U.S National Institute of Standards and Technology:

"Cloud computing is a model for enabling ubiquitous, 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 is composed of five essential characteristics, three service models, and four deployment models." [4].

3. SECURITY

The cloud computing did not bring only benefits, but also many threats. According to NIST, security, interoperability and portability are the key barriers to greater adoption of cloud.

Security issues of cloud computing the most discussed can be grouped into four major categories [9]:

- Cloud infrastructure: includes concerns about virtualization, storage and network vulnerabilities as well as the code and software hosted in the cloud computing, and the physical safety aspects of the data center.
- Data: includes the concerns about data integrity, availability and confidentiality and user privacy.
- Access: concerns around access to the cloud (authentication control, access authorization), encryption of communication, and management of user identity.
- Compliance: cloud must settle some issues concerning the regulation (security auditing, data localization and traceability).

It is necessary to meet the security requirements at each level in order to preserve data security in the cloud (confidentiality, integrity, availability and non-repudiation). Moreover, one must be sure of the effectiveness of these measures, their robustness, their resistance to attacks and their relevance to customer expectations and administrators Cloud.

Ten cloud computing obstacles were identified by a group of University of California at Berkeley research [8] (service availability, data privacy, blocking, software licenses ...).

The Cloud Security Alliance (CSA) identifies thirteen areas of concern on the security of cloud computing [10]. Data protection and confidentiality in the cloud is similar to traditional data protection and confidentiality. Security must be involved at every level of the data life cycle. Due to multi-tenancy, protection and confidentiality of data in the cloud become particular.

4. HOMOMORPHIC ENCRYPTION FOR DATA PRIVACY IN THE CLOUD

Among cloud computing characteristics, the sharing of conservation structures and data processing, one problem of this is the preservation of confidentiality between client and provider. Encryption could alleviate this issue, since the customer can decide to store only encrypted data. The problem is that while data can be sent to and from a cloud provider's data center in encrypted form, the servers that power a cloud can't do any work on it that way. So if the client wants to perform calculations on its data in the cloud, the secret key to decrypt the data should be shared with the provider. Sharing the key would allow the cloud provider access to the data. The answer to this problem is the homomorphic encryption. The client would provide the cloud with executable code to allow it to work on the data without decrypting it. The result will be returned to the client still encrypted. So since the client is the only holder of the secret key, no one else is able to decrypt neither data nor results.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Notation:

An encryption scheme has three components (KeyGen, Enc, Dec):

- KeyGen: the function that generates pair of keys (public key pk and secret key sk).
  $(pk, sk) \leftarrow KeyGen(S)$
- Enc: an encryption algorithm that takes the public key and the plain text to crypt *M* and gives the ciphertext.
  $c \leftarrow Enc_{pk}(M)$
- Dec: a decryption algorithm that takes the ciphertext *c* and the secret key and recovers the plain text *M*.
  $M \leftarrow Dec_{sk}(c)$

An encryption scheme is homomorphic if we can make calculations equally well on plaintext data and on the encrypted data, and having the same result.

A homomorphic encryption scheme has a forth component which is the function Eval: applies a function *f* to a ciphertext *c* using the public key $c^* \leftarrow Eval_{pk}(f, c)$. A homomorphic encryption scheme must check the following properties.

- Additive homomorphism (AH):

A homomorphic encryption is additive if, $Dec_{sk}\left(Enc_{pk}(M1) \oplus Enc_{pk}(M2)\right) = M1 \oplus M2$ (1)

- Multuplicative homomorphism (MH):

A homomorphic encryption is multiplicative if, $D_k\left(C_K(M1) \otimes C_K(M2)\right) = M1 \otimes M2$ (2)

An algorithm is called fully homomorphic if both properties are satisfied simultaneously.

Below are some examples of existing homomorphic cryptosystems:

- RSA(multiplicative)
- ElGamal (multiplicative)
- Paillier (additive)
- Gentry (additive & multiplicative)

4.1. RSA encryption scheme

RSA encryption scheme introduced by Rivest, Shamir and Adleman [2] has multiplicative homomorphism. Recall that RSA cryptosystem works like:

- To generate a public key/secret key pair, we choose two primes *p* and *q* and set *n* =*p.q*. we choose also an integer *e* coprime to $\phi(n) = (p-1)(q-1)$. The public key pk is *(n,e)* and the secret key sk is *(p,q)*. We note that given *p* and *q* it's easy to calculate $d = e^{-1} \bmod(\phi(n))$.
- An encryption of m is and the decryption is $Dec(sk, c) = c^d \bmod n$.

The homomorphic property is then:

$$\begin{aligned} Enc(p1).Enc(p2) &= p1^e\ p2^e \bmod n \\ &= (p1p2)^e \bmod n \\ &= Enc(p1.p2) \end{aligned}$$

This is saying if we take two plaintext messages p1 and p2 and multiply them together and then encrypt that using RSA, we get the same result as if we encrypt each plaintext separately and then multiply the two ciphertexts together.

4.2 ElGamal encryption scheme

ElGamal encryption algorithm is proposed by Taher ElGamal in 1984. It works as follows [1]:

- Let *p* a prime and *g* a generator. Pick *x* randomly from *{1... p-1}* and compute $h = g^x \bmod p$. The *p*, *h* and *g* are public and *x* is private.
- Let $r \in Z_{p-1}$ be a secret random number, then the encryption of the message *m* is $Enc(m, r) = (g^r \bmod p, m h^r \bmod p)$.

ElGamal has also a multiplicative homomorphic property. Given two plaintexts m1 and m2, the homomorphic property is then:

$$\begin{aligned} Enc(m1, r1).Enc(m2, r2) &= (g^{r1} \bmod p, m1\ h^{r1} \bmod p). \\ &\quad (g^{r2} \bmod p, m2\ h^{r2} \bmod p) \\ &= (g^{r1+r2} \bmod p, (m1\ m2)\ h^{r1+r2} \bmod p) \\ &= Enc(m1m2, r1\ r2) \end{aligned}$$

Journal of Theoretical and Applied Information Technology
20th December 2015. Vol.82. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.3 Paillier encryption scheme

The scheme works as follows [3]:

- Choose two large prime numbers p and q randomly and independently of each other and sets $n= pq$, such that $\gcd(n, \phi(n)) = 1$.
- Let $\lambda(n) = lcm(p-1, q-1)$, and pick g such that $1 \leq g \leq n^2$ and $L(g^{\lambda} mod\ n^2)$ is invertible modulo n. The public key pk is (n,g) and the secret key sk is (p,q,λ).
- An encryption of message $m < n$ is given by : $c = g^m\ r^n\ mod\ n^2$ and the decryption of the ciphertext c is given by : $m = \left(\frac{L(c^{\lambda} mod\ n^2)}{L(g^{\lambda} mod\ n^2)}\right) mod\ n$

Paillier follows additive homomorphic encryption. By multiplying each component of multiple ciphertexts with their corresponding respective components, the decrypt result is equivalent to the addition of the plaintext values.

Consider two plaintext messages p1 and p2 with corresponding ciphertexts:

$$Enc(p1, r1) = g^{p1} r1^n\ mod\ n^2$$
$$Enc(p2, r2) = g^{p2} r2^n\ mod\ n^2$$

Multiplying the two ciphertexts together yields:

$$Enc(p1, r1).Enc(p2, r2) = g^{p1} r1^n . g^{p2} r2^n\ mod\ n^2$$
$$= g^{p1+p2} (r1\ r2)^n\ mod\ n^2$$
$$= Enc(p1 + p2, r1r2)$$

4.4 Fully homomorphic encryption

Up until now, the homomorphic systems described have been partially homomorphic: they preserve the structures of multiplication or addition, but cannot do both. We talk about fully homomorphic encryption if any arbitrary computation could be performed on a ciphertext, preserving the encryption as if the computation was performed on the plaintext.

Boneh, Goh and Nissim [7] were the first to construct a scheme capable of performing both operations at the same time: their scheme handles an arbitrary number of additions but just one multiplication.

In 2009, Graig Gentry proposed the first fully homomorphic encryption (FHE) [5]. The use of fully homomorphic encryption is an important security solution in the cloud computing, particularly for those that wish to house encrypted data on cloud providers' servers.

5. PERFORMANCE

In the following we provide a comparison of partial homomorphic encryption and fully homomorphic encryption cryptosystems.

Table 1: Partial HE vs Fully HE

Partial HE	Fully HE
Allows either additive or multiplicative operations	Allows both additive and multiplicative operations
Key used by the client (different keys are used for encryption and decryption)	Key used by the client (different keys are used for encryption and decryption)
Security applied to cloud provider server	Security applied to cloud provider server
Requires less computational efforts	Requires more computational efforts
Privacy of data is ensured in both communication and storage process	Privacy of data is ensured in both communication and storage process
Faster in performance	Slower performance
Small ciphertext size	Large ciphertext

From the table, we can see clearly that speed and cipher size are the main problem for the fully homomorphic encryption scheme.

Performance of fully homomorphic encryption:

The implementation of Gentry's fully homomorphic encryption scheme is tested with the model of several dimensions, ranging between 2048 and 32768. The public-key size ranges in size from 70 Megabytes for the "small" setting to 2.3 Gigabytes for the "large" setting. Performance of fully homomorphic encryption with the parameters such as dimensions, key generation time, size of the public key is tabulated [1]:

Journal of Theoretical and Applied Information Technology
20th December 2015. Vol.82. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

	Dimension	Key Gen	PK size	AND
Small	2048 800,000-bit integers	40 sec	70 MByte	31 sec
Medium	8192 3,200,000-bit integers	8 min	285 MByte	3 min
Large	32768 13,000,000-bit integers	2 hours	2.3 GByte	30 min

6. IMPLEMENTATION

Encryption of data stored in the cloud has the advantage of enhancing the security measure of untrusted systems or applications that stores or manipulates sensitive data. The cloud computing scenario can be illustrated as below.

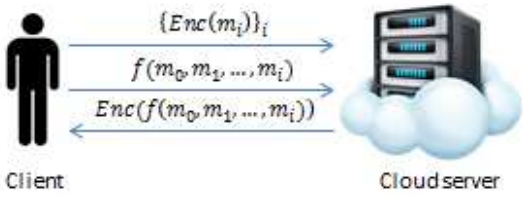

Figure 1: The Data Security Scheme For Cloud Computing

a) The client's application generates the public and private keys.

b) The application encrypts the data using the public key and sends it to the cloud server.

c) The client sends a request to the cloud for calculating the function $f(m_0,\ldots,m_i)$.

d) The cloud calculates the result of request sent by the client.

e) The cloud server computes $f(Enc(m_0),\ldots,Enc(m_i))$ without knowing $\{m_i\}_i$.

f) The cloud sends back the result of $f(Enc(m_0),\ldots,Enc(m_i))$ to the client.

g) The client can decrypt the encrypted result using the private key $Dec(f(Enc(m_0),\ldots,Enc(m_i)))$ and obtains the same result as if the calculation was carried out on raw data.

Example

Suppose that the company X has a very important data set that consists of the numbers 1 and 2. Company X encrypts the data so that 1 becomes 13 and 2 becomes 26. The company sends the encrypted set to the cloud for safe storage. Few months later, X needs to sum the numbers 1 and 2. The encrypted data is then processed: the result 39 (13+26 = 39) can be downloaded from the cloud and company X can decrypt it to provide the final answer 3.

7. CONCLUSION

In the age of the cloud computing, users move their sensitive data to the cloud. Trust must be placed in the provider's infrastructure to ensure data confidentiality. Even though the trusted provider, users and malicious programs can lead to data leakage.

Encryption is the solution to this problem, in addition to ensuring privacy stored data, allows to operate on them publicly without having to decipher them. This paper analyzes the application of different homomorphic encryption cryptosystems on a cloud computing platform.

Our work is based on the application of homomorphic encryption to the security of cloud computing, particularly the possibility to perform operations of encrypted data without decrypting them.

In future, we will focus on the development of a hybrid homomorphic encryption system based on existing partially homomorphic encryption algorithms applied to the medical domain.

REFRENCES:

[1] O. Ugus, A. Hessler and D. Westhoff, "Performance of additive homomorphic EC-Elgamal encryption for TinyPEDS," *Fachgespräch Sensornetzwerke,* pp. 55-58, 2007.

[2] R. L. Rivest, A. Shamir and L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

of the ACM, vol. 21, pp. 120-126, 1978.

[3] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," *Advances in Cryptology — EUROCRYPT '99,* vol. 1592, pp. 223-238, 1999.

[4] P. Mell and T. Grance, "The NIST definition of cloud computing," September 2011.

[5] C. Gentry and S. Halevi, "Implementing Gentry's Fully-Homomorphic Encryption Scheme," *Advances in Cryptology-EUROCRYPT,* vol. 6632 , pp. 129-148, 2011.

[6] C. Gentry, "A fully homomorphic encryption scheme," Stanford University, 2009.

[7] D. Boneh, E.-J. Goh and K. Nissim, "Evaluating 2-DNF Formulas on Ciphertexts," *Theory of Cryptography,* pp. 325-341, 2005.

[8] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, "Above the Clouds: A Berkeley View of Cloud," Electrical Engineering and Computer Sciences - University of California, Berkeley, 2009.

[9] A. Ait Elmrabti, A. Abou El Kalam and A. Ait Ouahman , "Les défis de Sécurité dans le Cloud Computing, Problèmes et solutions de la sécurité en Cloud Computing," 2012.

[10] Cloud Security Alliance, "Security guidance for critical areas of focus in cloud computing v3. 0," 2011.

[11] S. D. Tetali, M. Lesani and R. Majumdar, "MrCrypt: static analysis for secure cloud computations," in *Proceedings of the 2013 ACM SIGPLAN international conference on Object oriented programming systems languages & applications*, New York, 2013.