Journal of Theoretical and Applied Information Technology
29th February 2016 Vol 86. No.3

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

DATA CONFIDENTIALITY IN THE WORLD OF CLOUD

[1]KHALID EL MAKKAOUI, [2]ABDELLAH EZZATI, [3]ABDERRAHIM BENI-HSSANE, [4]CINA MOTAMED
[1,2] LAVETE laboratory, Department of Mathematics and Computer Science, Sciences and Techniques Faculty, Hassan 1st University, Settat, Morocco
[3] LAROSERI laboratory, Department of Computer Science, Sciences Faculty, Chouaïb Doukkali University, El Jadida, Morocco
[4] LISIC Laboratory, Littoral Cote d'Opale University, Calais, France
E-mail: [1]kh.elmakkaoui@gmail.com, [2]abdezzati@gmail.com, [3]abenihssane@yahoo.fr, [4]motamed@lisic.univ-littoral.fr

ABSTRACT

Cloud computing is becoming an attractive technology thanks to its diverse benefits such as: reducing costs, sharing computing resources, service flexibility, etc. However, the concept of data security and privacy has become a major issue. Indeed, the key challenge is to ensure users that the cloud service provider may store and process the raw data confidentially. The fear of seeing sensitive raw data stored and used are the major barrier to cloud services adoption. The use of methods capable of ensuring storage confidentiality and data processing located in cloud servers appears to be an effective way to overcome this barrier and to build confidence in cloud services. This paper describes a layered model of cloud security and privacy, and some approaches used for secure data in cloud environment. Specifically, we will focus on the encryption methods that can ensure data storage confidentiality and another technique that can ensure both data storage confidentiality and data treatments.

Keywords: *Cloud Computing, Security, Privacy, Confidentiality, Symmetric key, Homomorphic Encryption.*

1. INTRODUCTION

According to a definition given by the NIST (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". NIST defined five essential characteristics that distinguish cloud from other technologies, namely: on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service [1].

In other words, if a company subscribes to the cloud, distributed computing would be exactly like water, gas or electricity: it only has to connect to benefit from it. The bill is done according to consumption and during periods of non-use (holidays, vacations, etc.), simply close the meter. In general, cloud computing is composed of three basic service models: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) [1], and four deployment models: public (the physical infrastructure is owned and managed by the service provider), community (the physical infrastructure is owned and managed by a consortium of organizations), private (the infrastructure is owned and managed by a specific organization) and hybrid cloud (include combinations of the previous three deployment models) [1],[2].

Indeed, cloud computing is becoming more and more a magical solution thanks to its enormous benefits including reducing costs, sharing and configurable computing resources, computer park maintenance, and service flexibility. However, the cloud security has become a major matter of concern for users and organizations that want to adopt and benefit from cloud services. The adoption of these services is limited by concerns about the loss of privacy and the value of private data [4]. These concerns on the data confidentiality do not only exist when uploading and retrieving it to/from the cloud, but also when the data located in cloud servers of a non reliable cloud services provider [5].

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

Therefore, cloud providers must adopt techniques to preserve the confidentiality of data, in order to build confidence in the provided services.

In this regard, a number of research works have been done and several techniques are proposed to ensure the data security in cloud environment. In reference [6] PENG Yong et al. give a review on research results of secure cloud storage based on cryptographic techniques. In [7] the authors provide a survey on different cloud security issues and on different cryptographic algorithms, adoptable to secure cloud. In [5] the authors provide a brief overview of some encryption schemes used in cloud computing environment to secure data sharing. However, most of the existing research works give a general overview of some encryption methods used to secure cloud. In our work, we will provide a deep study on security and privacy of cloud issues, and we will discuss some encryption methods used to ensure data confidentiality in cloud environment. We will divide these encryption methods into two types: encryption methods used to ensure data storage confidentiality, and other encryption methods used to ensure data storage confidentiality and data treatments.

The rest of this paper is organized as follows: In Section II, we will present a layered model of cloud security and privacy. In Section III, we will present some encryption methods used to ensure the confidentiality of data storage. In Section IV, we will present Homomorphic Encryption technique that can able to ensure the confidentiality of storage and data treatments. Finally, we will finish with section V, in which we will present our conclusions and future work.

2. CLOUD SECURITY AND PRIVACY MODEL

In recent years, many researchers and organizations worked on identifying cloud security and privacy issues. In [8], Issa M. khalil et al. provided a deeper classification of cloud security related issues of the five categories: security standards, network, access control, cloud infrastructure and data. The Cloud Services Measurement Initiative Consortium (CSMIC) developed a standard measurement framework, called Service Measurement Index (SMI). This framework is designed to become a standard method to help organizations measure cloud-based business services, based on their specific business and technology requirements. The framework SMI is hierarchical and divided into 7 categories including: Accountability, Agility, Assurance, Financial, Performance, Security and Privacy, and Usability, and each category is refined by 3 or more attributes [9],[10]. The security and privacy category includes attributes namely: Access Control and Privilege Management, Data Geographic/ Political, Data Integrity, Data Privacy and Data Loss, Physical and Environmental Security, Proactive Threat and Vulnerability Management, Retention/Disposition, and Security Management [10].

According to reference [8], the security standards category includes service level agreements, auditing and other agreements among users, service provider and other stakeholders. Those define cloud security policies in order to ensure secure working over the cloud environment. But, CSMIC classified the principle of security standards category in accountability category of SMI framework.

In this section, we will propose a secure and private cloud model into layer, based on these works. This model is divided into five layers: Physical and Environmental Security, Cloud Infrastructure Security, Network Security, Data, and Access Control and Privilege Management. As shown in Figure 1.

This model can be considered as a secure and private map. Thus, it can help cloud providers to take into account the concept of security and privacy during all the stages of cloud services building, in order to provide secure services to cloud users. Also, it can allow the researchers to identify different layers of cloud security and privacy issues.

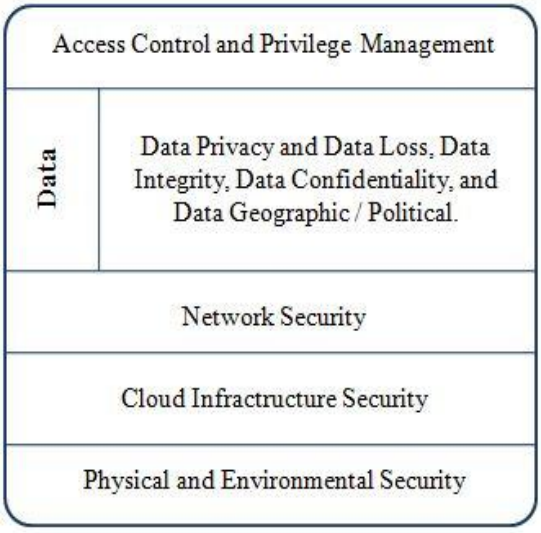

Figure 1. Cloud security and privacy model

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

Layer1: *Physical and Environmental Security*: Processes and policies adopted by the cloud services provider to protect their facilities against unauthorized physical access, damage, etc [10].

Layer2: *CloudInfrastructure Security*: Includes security issues specific to cloud infrastructure (IaaS, PaaS and SaaS) and is particularly related with virtualization environment. Security issues of this layer includes: insecure interface of API, sharing technical flaws, security misconfiguration, multi-tenancy, etc. [8],[11].

Layer3: *Network Security*: Refers to the medium through which the users connect to cloud services, including browsers, network connections, etc [8].

Layer4: *Data*: This layer is composed of Data Geographic / Political, Data confidentiality, Data Integrity, and Data Privacy and Data Loss.

- *Data Geographic / Political*: Constraints of the client on the location of services, based on geographical or political risk [10].
- *Data Confidentiality*: Ensures that data remains confidential and invisible even to the cloud provider, and even if the provider data centre have been attacked, customer data can neither be stolen nor reused [5],[12],[13].
- *Data Integrity*: Keeping data in its correct form. It means that the system must prevent undue modification of information (i.e. a modification by unauthorized users or incorrect modification by authorized users) [10],[13].
- *Data Privacy and Data Loss*: The cloud services provider enforce the restrictions on clients to use and share data. Any failures of these protections are rapidly detected and reported to the client service [10].

Layer5: *Access control and Privilege Management*: Policies and processes used by cloud services provider to ensure that only the users granted appropriate privileges can use or modify data. It includes identification, authentication and authorization issues [8],[10],[11].

In the rest of this paper, we will focus on the layer 4 of cloud security and privacy model, and particularly on: Data Confidentiality. We will present some techniques and encryption methods used to ensure the confidentiality of data storage and treatments in cloud environment.

3. CONFIDENTIALITY OF DATA STORAGE

Cryptography algorithms are the most efficient tool to ensure the security of data storage in the cloud. Indeed, there are many encryption algorithms that can encrypt the data and convert them into incomprehensible format, in order to ensure their confidentiality. These algorithms are divided into two categories: symmetric and asymmetric key [14] (as shown in the figure 2). The asymmetric key techniques performance is very slower than the symmetric key techniques [7], and used in general to exchange the keys of symmetric key algorithms [14]. The symmetric key algorithms are a form of encryption that use same key to encrypt and decrypt the data, and are divided into block ciphers and steam ciphers [7], [15]. The input of block cipher when the data is encrypted or decrypted is in block of the fixed data size. This size is depending on the encryption algorithm used. But, in the case of stream cipher the data are encrypted/ decrypted one bit or one byte of data at time [14], [7], [15], [16] that's why this cipher type is more efficient for real time processing [17]. In this section, we focus on symmetric key algorithms which can be adopted by cloud providers to ensure the confidentiality of the data storage.

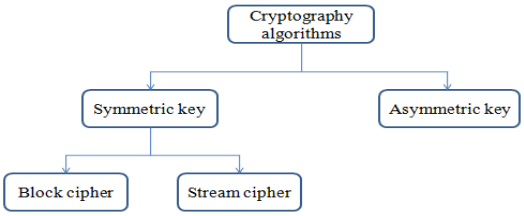

Figure 2. Categories of cryptography algorithms

3.1 Principles of Symmetric Key Cryptosystems Operation

The philosophy of symmetric key cryptosystems when adopted to encrypt data in the cloud environment is somewhat different to other use cases. Generally, symmetric key techniques were created to encrypt messages, to ensure confidentiality of communication between the sender and the recipient over internet and network applications. The secret key is shared between them

Journal of Theoretical and Applied Information Technology
29th February 2016 Vol 86. No.3

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

to encrypt and decrypt these messages. In cloud case these cryptosystems are used to ensure confidentiality of data, when transferred and when stored in cloud servers. Only the client can have the key. Thus, the data remains confidential and unreadable even to the cloud provider.

The principles of operation of the symmetric key cryptosystems in cloud, are as follows, and as shown in Figure3.

- **Key Generation**: The client generates private key (**PK**).
- **Encryption**: The client encrypts data with P**K** and sends the encrypted data to the Cloud server.
- **Storage**: The encrypted data are stored in the cloud database.
- **Decryption**: The client decrypts data after retrieve them from the cloud, using **PK**.

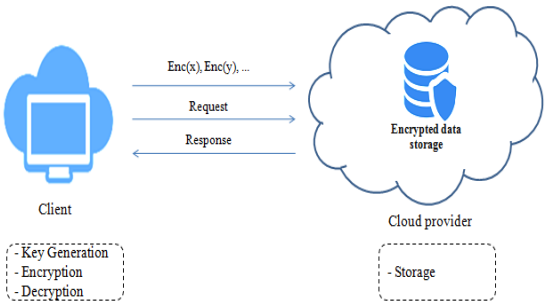

Figure 3. Functioning principle of symmetric key cryptosystems

3.2 Symmetric Key Cryptosystems

The Symmetric key cryptosystems are numerous. Among them, we will present two cryptosystems: Blowfish and RC4. The selection of these two cryptosystems was based on some research works. In reference [14] authors provide a comparison between four symmetric key algorithms (DES, 3DES, AES and Blowfish) and they evaluate their performances. The comparison is done during encryption and decryption time, CPU process time in the form of throughput, and power consumption. Among these algorithms, Blowfish has the best performance (the more the speed of encryption and decryption of data is, the less the consumption of the battery will be). In [17] authors compare two algorithms: AES (block cipher) and RC4 (stream cipher), on ground of encryption/decryption time, throughput, CPU process time and memory utilization. They concluded that RC4 is faster and better than AES in term of energy efficient for encryption and decryption. And in [15] H.Agrawal and M. Sharma provide a benchmarking between some of the widely used symmetric encryption techniques (DES, 3DES, AES, Blowfish and RC4), and they concluded that Blowfish (block cipher) and RC4 (stream cipher) have the best performance regarding the memory required for implementation and the time required by the algorithms for data processing.

3.2.1 Blowfish algorithm

Blowfish is a symmetric key block cipher; it was developed by Bruce Schneier in 1993. The block size is 64 bits, and the key can be any length between 32 and 448 bits (128 bits by default). It is a Feistel network, and its algorithm consists of two parts: a key expansion and a data encryption. Key expansion converts a key at most 448 bits into several arrays totaling 4168 bytes. Data encryption occurs via a 16 rounds, each round consists of a key-dependent permutation and a key data-dependent substitution, and all operations are XORs and additions on 32 bit words [18]. Blowfish is a robust algorithm; it encrypts data on 32 bit of microprocessors and it can run in less than 5K of memory [15]. The figure below represents blowfish cryptosystem.

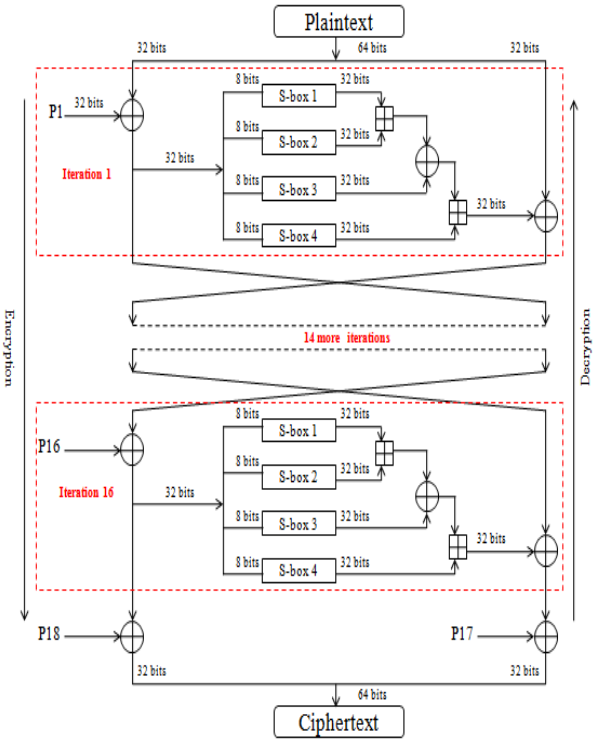

Figure 4. Blowfish cryptosystem

Journal of Theoretical and Applied Information Technology
29th February 2016 Vol 86. No.3

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

3.2.2 RC4 algorithm

RC4 (Rivest Cipher 4) is a symmetric key stream cipher designed in 1987 by Ron Rivest [17]. This algorithm is used in popular protocols such as: TLS, WEP and WPA [6]. RC4 encrypts data one byte at a time. It can use any length key from 1 to 256 bytes [19]. Its algorithm is based on the use of a random permutation [17], which consists of two phases (as shown in the figure 5): the key scheduling and pseudo random generator [19]. The key scheduling generates the initial permutation from the key. And pseudo random generator produces one byte output in each step. The encryption is an XOR of the pseudo random sequence with byte of the plaintext stream, and the decryption is an XOR of the pseudo random sequence with byte of the ciphertext stream (as shown in figure 6).

Key scheduling

1. Initialization:
   for i = 0 **to** 255
     S[i] = i
   end for
   j = 0
2. Generate a random permutation:
   for i = 0 **to** 255
     j= (j + S[i] + K[i **mod** length_key]) **mod** 256
     swap (S[i], S[j])
   end for

Pseudo random generator

1. Initialization:
   i = 0
   j = 0
2. Generate pseudo random sequence:
   loop
     i = (i + 1) mod 256
     j = (j + S[i]) mod 256
     swap (S[i], S[j])
     t = (S[i] + S[j]) mod 256
       print S[t]
   end loop

Figure 5. RC4 key scheduling/ pseudo random generator algorithms

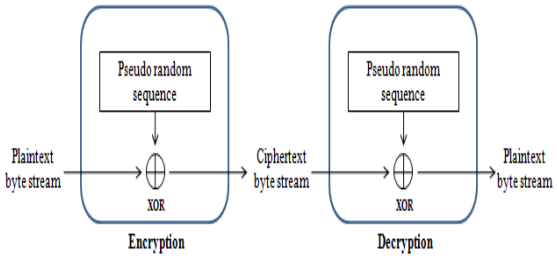

Figure 6. RC4 encryption / decryption algorithms

In the table blew presents the some popular symmetric key algorithms.

Table 1: Symmetric Key Algorithms.

Algorithms	Cipher type	Key Size	Algorithm Structure	Attacks
DES	Block (64 bits)	56 bits	Fiestel Network	Brute force attack
3DES	Block (64 bits)	K1,k2,k3 168 bits	Fiestel Network	Theoretically possible
RC4	Stream	8- 2048 bits	Fiestel Network	Fluhrer mantin and shamir attack
Blowfish	Block (64 bits)	32-448 bits (128 by default)	Fiestel Network	Differential Attack , Weak key
AES	Block (128, 192 or 256 bits)	128,192 or 256 bits	Substitution Permutation Network	Side channel attacks
Twofish	Block (128 bits)	128, 192 or 256 bits	Fiestel Network	Truncated differential cryptanalysis

4. CONFIDENTIALITY OF DATA STORAGE AND DATA TREATMENTS

In an insecure environment like the public cloud, sensitive data must be secured. Regarding the storage service, data can be encrypted before sending them to the cloud server, using the symmetric key cryptosystems such as: Blowfish, etc. But, to ensure the confidentiality of data storage and their treatments, the cloud providers must adopt techniques that can ensure the confidentiality of this type of service. Indeed, researchers stressed a useful encryption technique in this type of environment: Homomorphic Encryption (HE). This technique is able to ensure the confidentiality of data storage and their treatments, located in cloud servers [3], [4], [20], [21]. The Homomorphic Encryption cryptosystems are asymmetric key, which use different keys for data encryption and decryption. In this section, we will define first Homomorphic Encryption technique and secondly introduce different operations types. In the third part we will see the categories that compose it, and finally present some of HE cryptosystems.

Journal of Theoretical and Applied Information Technology
29th February 2016 Vol 86. No.3

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

4.1 Definition

Mathematically, we say that an encryption system is homomorphic if: from Enc(x) and Enc(y), it is possible to calculate Enc(f(x, y)), without using the private key, where f can be: +, ×, ⊕ [22].

In other words, Homomorphic Encryption systems are capable of performing operations on encrypted data without knowing the private key. These operations generate a result, which is itself encrypted (i.e. incomprehensible even to cloud provider). The result obtained is the same as if we performed these operations on raw data [3], [20], [21].

4.2 Principles of Operation

The principles of operation of the Homomorphic Encryption systems are as follows, and as shown in Figure7 [3], [20], [21].

- **Key Generation**: The client generates a public key (**pk**) and a private key (**sk**).
- **Encryption**: The client encrypts data with encryption key (**pk** or **pk**+**sk**). And sends the encrypted data and **pk** to the cloud server.
- **Storage**: The encrypted data and **pk**, are stored in the cloud database.
- **Request**: The client sends a request to the server to perform operations on encrypted data.
- **Evaluation**: The processing server processes the request and performs the operations requested by the client.
- **Response**: Cloud provider returns to the client the processed result.
- **Decryption**: The client decrypts the returned result, using **sk**.

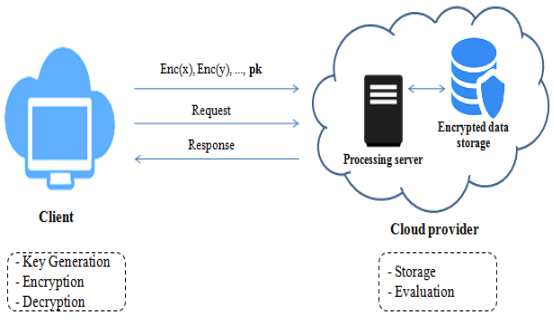

Figure 7. functioning principle of HE

4.3 Operations Types and Categories of HE

Among the Homomorphic Encryption systems we distinguish, depending on the operations which can evaluate the raw data, multiplicative and additive Homomorphic Encryption.

- *Multiplicative Homomorphic Encryption:* A Homomorphic Encryption is multiplicative, if there is an algorithm which can calculate Enc (x × y) from Enc(x) and Enc(y) without knowing x and y [23].
- *Additive Homomorphic Encryption:* A Homomorphic Encryption is additive, if there is an algorithm whitch can calculate Enc (x + y) from Enc(x) and Enc(y) without knowing x and y [23].

Among Homomorphic Encryption systems, we distinguish three categories, depending on the operations performed on the data :

- **Partially Homomorphic Encryption (PHE):** Allows to perform operations on encrypted data such as multiplication or addition, but not both [24].
- **Somewhat Homomorphic Encryption (SWHE):** Allows to perform more than one operation, but a limited number of multiplication and addition operations [24].
- **Fully Homomorphic Encryption (FHE):** This is a cryptographic system that supports an unlimited number of both additions and multiplications [24].

4.4 Homomorphic Encryption Cryptosystems

Homomorphic Encryption cryptosystems are numerous. We will present the algorithms of three cryptosystems: GM [25], ElGamal [26] and EHES [27].

4.4.1 GM cryptosystem

The Goldwasser-Micali(GM) system is a probabilistic public-key encryption scheme, developed by Shafi Goldwasser and Silvio Micali in 1982. It is an additive Homomorphic Encryption, but it can encrypt just a single bit [25],[22]. GM algorithm as shown in the figure 8.

Journal of Theoretical and Applied Information Technology
29th February 2016 Vol 86. No.3

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

Preparation of Key
Input: p, q $\in \mathbb{P}$, p≠q (practically larger)
• Compute **n** = p×q • Choose **z** ∈ Z_n , such that **z**: a quadratic non-residue modulo **n** and $(\frac{z}{n}) = 1$
Output: (pk, sk) public key: **pk** = (n, z) secret key: **sk** = (p, q)

Encryption: Enc (m, pk)
Input: message **m** composed of **t** bits , $m_1m_2...m_t$
• Choose randomly: ∀ i ∈ [1,t] : r_i • Compute ∀ i ∈ [1,t] : $c_i \equiv z^{m_i} \times r_i^2 \bmod n$
Output: encrypted message **c** = $(c_1c_2...c_t)$

Decryption: Dec (c, sk)
Input: **c** = $(c_1c_2...c_t)$
• Compute $e_i = (\frac{c_i}{p})$, ∀ i ∈ [1,t] • If $e_i = 1$ so $m_i = 0$, else $m_i=1$
Output: **m** = $(m_1m_2...m_t)$

Figure 8. GM Algorithm

Suppose we have two bits to encrypt **m1** and **m2** by the GM cryptosystem.

Additive:

$$\text{Enc}(m_1)\times\text{Enc}(m_2) \equiv (z^{m1}\times r_1^2)(z^{m2}\times r_2^2) \bmod n$$
$$\equiv z^{m1+m2}(r_1r_2)^2 \bmod n$$
$$\equiv \text{Enc}(m_1 \oplus m_2, pk)$$

4.4.2 ElGamal cryptosystem

In 1984, Taher ElGamal proposed a public-key cryptosystem [26], which is a multiplicative Homomorphic Encryption system. The figure bellow represents their algorithm.

Preparation of Key
Input: **p** ∈ $\mathbb{P}$ (random and large prime)
• Find a generator g of the multiplicative group Z^*_p • Choose a random integer **a** ∈ [2, p-2] • Compute **y** = $g^a \bmod p$
Output: (pk, sk) public key: **pk** = (p, g, y) secret key: **sk** = (a)

Encryption: Enc (m, pk)
Input: message **m** ∈ Z_p
• Choose a random integer **k** ∈ [2, p-2] • Compute $c_1 \equiv g^k \bmod p$ $c_2 \equiv m \times y^k \bmod p$
Output: encrypted message **c** = (c_1, c_2)

Decryption: Dec(c, sk)
Input: **c** = (c_1, c_2)
Compute **m** ≡ $c_1^{-a} \times c_2 \bmod p$
Output: clear message **m** ∈ Z_p

Figure 9. ElGamal Algorithm

Suppose we have two encrypted messages **C1** and **C2** by the ElGamal algorithm, such as: $C_1=(c_{11}, c_{12})$ et $C_2=(c_{21}, c_{22})$.

Multiplicative:

$$(c_{11}, c_{12}).(c_{21}, c_{22}) \equiv (c_{11}c_{21}, c_{12}c_{22})$$
$$\equiv (g^{k1}g^{k2}, (m_1\times y^{k1})(m_2\times y^{k2}) \bmod p$$
$$\equiv (g^{k1+k2}, (m_1\times m_2)y^{k1+k2}) \bmod p$$
$$\equiv \text{Enc}(m_1\times m_2, pk)$$

4.4.3 EHES cryptosystem

In 2013, Gorti VNKV Subba Rao proposed Enhanced homomorphic Encryption Scheme (EHES) for homomorphic encryption / decryption with the IND-CCA secure system. This system allows to perform operations of addition, multiplication and mixed [27]. Their algorithm represents in the figure 9.

Preparation of Key
Input: p, q ∈ $\mathbb{P}$ (large prime numbers) , such that q < p
Compute : **n** = p×q
Output: (pk, sk) public key : pk = (n) Secret key : sk = (p, q)

Encryption : Enc (m, pk, sk)
Input: m ∈ Z_p
• Generate a random number **r** • Compute : c ≡ m + r × p^q (mod n)
Output: c ∈ Z_n

Decryption : Dec (c, sk)
Input: c ∈ Z_n
Compute : m ≡ c mod p
Output: m ∈ Z_p

Figure 10. EHES Algorithm

Let **x**, **y** ∈ Z_p, pk = (n) and sk =(p, q)

Multiplicative:

$$\text{Enc}(x \times y) \equiv (\text{Enc}(x) \times \text{Enc}(y)) \pmod n, \text{ or}$$
$$x \times y = \text{Dec}(\text{Enc}(x) \times \text{Enc}(y))$$
$$\equiv (\text{Enc}(x) \times \text{Enc}(y)) \pmod p$$

Additive:

$$\text{Enc}(x + y) \equiv \text{Enc}(x) + \text{Enc}(y) \pmod n, \text{ or}$$
$$x + y = \text{Dec}(\text{Enc}(x) + \text{Enc}(y))$$
$$\equiv (\text{Enc}(x) + \text{Enc}(y)) \pmod p$$