Journal of Computer Sciences

Original Research Paper

A New Authentication and Homomorphic Encryption as a Service Model for Preserving Privacy in Clouds

Karim Zkik, Maha Tebaa, Tarik Tachihante and Ghizlane Orhanou

Laboratory of Mathematics, Computing and Applications,
Faculty of Sciences, Mohammed V University in Rabat, Rabat, Morocco

Article history
Received: 29-08-2017
Revised: 15-09-2017
Accepted: 14-11-2017

Corresponding Author:
Karim Zkik
Laboratory of Mathematics, Computing and Applications,
Faculty of Sciences,
Mohammed V University in Rabat, Rabat, Morocco
Email: karim.zkik@gmail.com

Abstract: The security as a Service (SECaaS) is a new model which provides security solution to users through Cloud Computing. The maturity of Cloud Computing services makes possible the use of the SECaaS model. This new model offers huge benefits to users, such as Authentication as a Service (AaaS) and Encryption as a Service (ENCaaS). So, it can offer more security features, since it uses the resources of Clouds and it's connected to the different security policy databases. While SECaaS offers to cloud users and companies a multitude of security services, it still remains very limited and several aspects of security are not covered by this model, especially the part concerning the privacy. In addition, SECaaS is a new model that is not yet correctly deployed and it is not sufficiently solicited by companies. On the other side, Homomorphic encryption is considered as a good solution to ensure the privacy for users using the cloud services because it permits to make calculation on cipher text and data without decrypting them, but this solution suffer from many limitations such as the key size, the high latency and some serious performance problems. The main idea of this paper it's to propose a new security model to preserving user's privacy using homomorphic encryption while bypassing its limitations. So, This paper proposes a framework for Authentication and Homomorphic Encryption (A-HEaaS) based on security as a Service model which permits a secure access to the Cloud servers and the use of homomorphic encryption for calculations on encrypted data. The paper describes the design of our model and gives an implementation of our framework on medical Data.

Keywords: Security as a Service, Cloud Services, Homomorphic Encryption, Authentication and Identity Access Management, Confidentiality and Privacy

Introduction

Cloud computing provides users with an infinity of resources such as on-demand storage and calculation power and allows users to benefit from multiple services such as Software as a Service (SaaS), Infrastructure as a Service (IaaS) And Platform as a Service (PaaS) (Subashini and Kavitha, 2010).

Security as a Service (SECaaS) (Furfaro *et al*., 2014) is a new model that uses the resources of Cloud Computing to offer security services to users and companies. So, it allows among others to ensure the authentication and the security of personal data of Cloud users and offers safe solutions against intrusions and malwares. The security as a Service model is linked to Cloud services and to multiple databases, allowing users and companies to update the latest security policies and to protect themselves against the latest attacks and threats.

The SECaaS model has allowed opening new horizons for security experts who want to benefit from the diverse advantages of the Cloud. Several researches (Khan, 2016; Samlinson and Usha, 2013) have been based on the Security as a Service model and several architectures were proposed to offer to the users various applications and security solutions issues such as authentication, confidentiality, privacy and integrity.

The SECaaS is a new model that offers a range of IT security services to Cloud users (Alliance and Secaa, 2011). However, this model is not yet sufficiently mature and does not include all aspects of computer security and especially privacy. Since the use of the SECaaS model

requires the externalization of data and operations related to the security field such as authentication and encryption; users and businesses using this type of service will face a major challenge and will lose automatically control on their data. In addition, several operations require computing on encrypted data, which forces companies to share their encryption keys with the cloud service provider, causing a serious security breach and putting them in front of a real danger.

The SECaaS model also suffers from several limitations in its implementation because it relies on a client/server model with several entry and outputs points necessary for communication with the end users. These multiple points are used primarily for sending and receiving multiple requests. So, they can be used by malicious people, who want to intercept and make sniffing on these various communications, so they can be considered as points of failure.

There are also several other challenges that require our attention when using this type of cloud service (Delamore and Ko, 2015; Getov, 2012). For example during deployment of SECaaS, the various existing services are generally used in a parallel and uniform manner, so that once there is a security breach on one service or request, security is broken for all the system.

On the other hand, the homomorphic encryption (Rivest, 2002) is a mathematical model that allows making calculation on encrypted data without decrypting them. This mechanism is considered as a great asset which allows to ensures the users' privacy, but it still suffers from several limitations such as the encryption keys size and the huge calculation time that it consumes, which makes it almost unusable.

The basic idea of this article is to develop a new model of Authentication and Homomorphic Encryption as a Service (A-HEaaS) based on the SECaaS model. This will allow us to use Cloud resources to overcome the limitation of homomorphic encryption and to be able to make calculations on encrypted data. Thus, we will design an architecture that consists of three mechanisms:

- A secure authentication mechanism to allow users to connect to the remote public cloud servers safely
- A homomorphic encryption mechanism that allows encryption by using some encryption systems with homomorphic properties
- A homomorphic computational mechanism that permits to make calculations on encrypted data without decrypting them

Through this work we will try to answer several questions and address several issues regarding encryption key management, performance, latency and the establishment and deployment of a new authentication and homomorphic encryption as a services model to manage identity access and to make calculations on encrypted data. This will allow us to propose a solution to the requirements of cloud users and companies in relation to the confidentiality and security of their data and to the preservation of their privacy. We also propose in this work a new method of implementation of the homomorphic encryption based on the Cloud resources to exceed its limitations.

The paper is structured as follows. In section 2 we discuss some related works on security as a service model, authentication as a service model and on homomorphic encryption. In section 3, we present our proposed framework and we present a global view of our architecture by giving an overview of each of our authentication, homomorphic encryption and calculation mechanisms. We perform also an implementation of our A-HEaaS model to prove its efficiency. In section 4 we conclude the paper.

Theoretical Background and Research Scope

In this section we will define several theoretical concepts such as SECaaS and homomorphic encryption since they represent the core of our work. We will then present some of their related works and discuss their contributions and limitations. This will then allow us to define the different axes on which we will work and the challenges that will be addressed.

Background

Security as a Service Model

Security-as-a-Service (SaaS) is a new model for security management who includes a wide range of security services and involves applications such as anti-malwares software, intrusion prevention system, authentication mechanism and encryption/decryption processes. Those services are delivered over the internet and provided generally by an external organization.

The concept of Security as a Service was introduced in the last recent years and several studies have been conducted to define and improve this new model. The Cloud Security Alliance (CSA) (Alliance and Secaa, 2011) publishes a manuscript on SECaaS and they define Security as a service under 10 categories:

- Identity and Access Management
- Data Loss Prevention
- Web Security
- Email Security
- Security Assessments
- Intrusion Management
- Security Information and Event Management (SIEM)
- Encryption
- Business Continuity and Disaster Recovery

- Network Security

Getov (2012) gives an overview on Cloud security services, discuss the different opportunities and concerns of security as a service and specifies that Cloud can provide four security mechanisms to users:

- Email Filtering
- Web Content Filtering
- Vulnerability Management
- Identity-as-a-Service (IDaaS)

According to MarketsandMarkets (Marketsandmarkets.com, 2016). The SECaaS market is already worth more than 3 billion dollars and expected to surpass 8 million dollars by 2020. According to that many Cloud providers, organization and vendors are interesting by this model. Many organizations have no adequate knowledge on security and they do not have sufficient resources to manage their own data. So, these companies paying more and more attention and interest to this new range of security services model because the use of the SECaaS offers a number of benefits, including:

- It allows users to be connected to multiple databases and receive the latest signatures updates of different malwares, so they will be much better protected
- The use of SECaaS enables companies to benefit from the expertise of security experts and benefit from greater expertise greater than is typically available within the organization
- Outsourcing of administrative tasks, such as log management which permit to save time and money
- Companies will no longer need to invest in safety equipment, which will allow them to devote more time to affine their core competencies

Despite the different SECaaS advantages, this model suffers from several problems related to its deployment. The main challenges of SECaaS are as follow:

- The problem of privacy can be seen as one of the main challenges of SECaaS. Several organizations refuse to use the services of this model because it does not on in any case guarantee their privacy
- Encryption can be an adequate solution to this problem, but we still need to decrypt the data if we want to make calculations on these encrypted data. In other words, to take full advantage of the resources of the Cloud by outsourcing the calculations, it is imperative to outsource the encryption and decryption keys, which can cause a serious security problem
- The SECaaS is based on a Client / Server model, so to authenticate users and to encrypt and filter data, we need a mechanism for sending and receiving data. In other words we must establish secure communication channels between cloud servers and end users to secure queries and different transmissions. A bad implementation can be used by hackers to sniff, to listen and to compromise sensitive user's data
- The SECaaS makes all security handling uniform so that once there is a security breach for one request; security is broken for all requests. Several attackers try to find the weak link in the architecture in order to access the whole system. This issue can cause serious damage to security service providers and to all users who use these services

In this study we will introduce the main works on SECaaS area and we will propose a new model based on SECaaS to deal whit all this challenges and we propose a new framework which proposes a solution for privacy problem using the homomorphic encryption.

Somewhat/Fully Homomorphic Encryption

The purpose of homomorphic encryption is to allow computation on encrypted data (Rivest, 2002). This mathematical concept has opened huge opportunities in terms of security and confidentiality of data and it allows offering more privacy to users who use the services of Cloud Computing.

The use of homomorphic encryption suffers from many limitations such as calculation time, calculation resources need and key sizes.

The calculation on data can be divided on two types: Addition and multiplication. Several cryptosystem are homomorphic, but these cryptosystems are still limited because they can't make addition and multiplication simultaneously. For example RSA is only homomorphic for multiplication and Pailler is only homomorphic for addition.

To mathematically explain the functioning of homomorphic encryption, we proceed as follows:

Assume that we have two values m_1 and m_2 and $E(m_1)$ and $E(m_2)$ are their respective encrypted function values, the function E in homomorphic encryption schemes is typically restricted to be an algebraic operation associated with the structure of the plain texts. Let's the plaintext space is a group G, then the cipher text space for homomorphic encryption schemes for multiplication is the product $G \otimes G$ and the cipher text space for homomorphic encryption schemes for addition is $G \oplus G$. The homomorphic encryption calculation is as follow:

$$E:G \to G \otimes G$$
$$m_1 \otimes m_2 = C \Rightarrow E(m_1) \otimes E(m_2) \to E(C) \quad (1)$$

$$E:G \to G \oplus G$$
$$m_1 \oplus m_2 = C \Rightarrow E(m_1) \oplus E(m_2) \to E(C) \quad (2)$$

In the standard homomorphic encryption schemes, the function E is restricted to the group operation on *G*.

Fully Homomorphic Encryption (FHE) allows making addition and multiplication simultaneously on encrypted data. We can express the objective of fully homomorphic encryption to extend the function *E* to any function.

Gentry's works shows for the first time that FHE is theoretically possible (Gentry, 2009a; 2009b). His construction consisted of three parts:

- Firstly, construct an encryption scheme that is somewhat homomorphic. In other words, an encryption system which can make addition and multiplication at the same time on functions not very complex
- Secondly, simplify the decryption of this scheme as much as possible to reduce calculations. This operation is called squashing
- Finally, evaluate this simplified decryption function homomorphically to obtain cipher texts on which we can make calculation without decrypting them

While the huge advantages of homomorphic encryption, efficiency was not the first priority in obtaining the first FHE schemes. It is theoretically possible to use Somewhat Homomorphic Encryption schemes to handle complex functions in using module 2 iterations. But the implementation of this system is very difficult because of the enormous number of operations it may require.

Several research studies focus on the problem of performance (Hemalatha and Manickachezian, 2014; Amna, 2013). Our work aims to propose a homomorphic computation scheme that uses standard homomorphic encryption schemes and Somewhat/Fully Homomorphic Encryption schemes (in our case we will use the encryption scheme DHGV). The aim of this work is to propose an efficient and usable homomorphic encryption and computation scheme for cloud users and the various companies that want to outsource calculation on their sensitive data. This model is based on Cloud Resource to bypass performance issues while maintaining a good level of security and ensuring the confidentiality and privacy.

Related Works and Research Scope

As we have explained before security as a service is a new model of services generally dedicated to Cloud users. This new concept allows companies to benefit from several advantages in terms of management and performance. In spite of this, SECaaS suffers from several limitations linked to its deployment and it faces several challenges. Several studies have addressed these issues and proposed several security architectures related to the proper use of this model (Rajkumar, 2014; Elsayed and Zulkernine, 2016):

Gupta and Gedam (2014) proposes one of the first security services model to counteract different attacks. Their architecture can be used by cloud provider to ensure security to their clients. Their model is composed from three security layer and at each level authentication was performed. So, only authorized client can get access to data or file stored at cloud.

Varadharajan (2014) proposes a security architecture that provides a flexible security as a service model that a cloud provider can offer to their users. So, it proposed a threat model that can protect the cloud infrastructure and provides additional security functionalities to users. In addition, he describes how different types of attacks could be counteracted by the proposed architecture.

Identity, Authentication and Access Management is one of the most important concerns in security.

So, the Cloud Security Alliance (CSA) (Alliance and Secaa, 2012a) offers a SecaaS Implementation Guidance for Identity and Access Management. The guidance discusses the reference architecture for Identity, Authentication and Access Management and discusses the significant technical decisions that need to be considered by an organization seeking to implement the IAM component of Security as a Service (SecaaS) as part of the cloud environment. The IAM components include:

- Centralized Directory Services
- Access Management Services
- Identity Management Services
- Identity Federation Services
- Role-Based Access Control Services
- User Access Certification Services
- Privileged User and Access Management
- Separation of Duties Services
- Identity and Access Reporting Services

Sharma *et al*. (2016a) proposes the Identity and Access Management as a Service (IAMaaS) framework. The IAMaaS architecture uses the specificities of SECaaS and it focuses on authentication, authorization, administration of Identities and audit. The framework aims to offers a good level of access to the cloud services and to verify users' identities. In particular, the IAMaaS is an on-demand portable and available pay-per-use cost model.

The confidentiality of data can be considered the most security concerns for users and security administrators. So, many researches and studies were made to offer a good level of data confidentiality.

Those frameworks are limited by the lack of resources, but the SECaaS offer a solution to all this limitations. So, Cloud Security Alliance (CSA) (Alliance and Secaa, 2012b) offers a SecaaS Implementation Guidance for encryption. The guidance discusses the reference architecture for encryption and security of data at all levels (data at rest, data in transit and data in use) on SECaaS Model.

Rahmani *et al*. (2013) developed an encryption as a service model based on private cloud to ensure confidentiality to Cloud users. The authors design an Encryption as a Service model in order to provide a solution to contract the security risks of cloud provider's encryption and to offer an alternative to the inefficiency of client-side encryption. The proposed model proposes also a key management process and its support many encryption schemes.

Sharma *et al*. (2016b) propose an Intelligent Transparent Encryption-Decryption as Security-as-a-Service model from Clouds and build a prototype of Encryption Decryption as-a-service (ENCaaS) model with an implementation and an evaluation.

In other hand, homomorphic encryption was introduced by Rivest (2002) on "Voting: Homomorphic Encryption" and it's discussed the different proprieties of the mathematical concept. As already mentioned, the implementation of the homomorphic encryption is not yet possible because of some problems related to performance. Several researches trying to find solutions and alternatives that will remedy to these problems while providing models that ensures users privacy in clouds (Wüller *et al*., 2017; Chen *et al*., 2014).

Tebaa *et al*. (2015) propose a hybrid homomorphic system that is applied to the banking data to perform operations on encrypted data without decrypting it. The idea of this work is to apply homomorphic encryption to very sensitive and very small data in order to increase efficiency and performance and to reduce computing time.

Poteya *et al*. (2016) proposes a framework to encrypt data using a homomorphic encryption scheme. To implement their model they use an amazon web server (DynamoDB). The purpose of their work is to ensure confidentiality and privacy to cloud users. The proposed framework can be used in many areas such as medical field and business purposes.

Hayward and Chiang (2015) proposes a framework that separates the data into several parts to process the encrypted data by fully homomorphic cryptosystems. This work aims to reduce computational loads and increase performance by using parallel processing on encrypted data. The authors used a client/server model based on the Gentry algorithm (Gentry, 2009b) when implementing their architecture.

As mentioned in the literature, there are several security architecture based on SECaaS model, but they do not guarantee privacy and suffer from several limitations related to their implementation. In contrast, solutions based on the use of homomorphic encryption also suffer from several problems in terms of performance management and computing time.

Based on the different results of these researches analysis we will build a SECaaS model that offer authentication and homomorphic encryption services for preserving privacy with enhancing performance and efficiently.

Authentication and Homomorphic Encryption as a SERVICE Model (A-HEaaS)

Global View of A-HEaaS

Ensuring the authentication and users' privacy is one of the major concerns of security experts (Werner *et al*., 2017; Zkik *et al*., 2017). The use of public cloud servers allowed offering several benefits to users in terms of resources but its limited by risks related to security, especially regarding confidentiality and privacy.

The homomorphic encryption provides an elegant solution that ensures data confidentiality and privacy. So, the data will never be decrypted, even by an administrator of the Cloud services provider.

Despite the apparent benefits of homomorphic encryption it is rarely used because it requires a very large encryption keys and an enormous calculations time especially on large data. To make the use of homomorphic encryption possible we will use the resources of the Cloud. So we will build a new model that offers authentication and homomorphic encryption services to users. Figure 1 shows the different components of our model.

To access to the remote cloud servers, the user must authenticate in a web application linked to our private Core Cloud. After the authentication, the user could interact with the Private Core Cloud to send an authentication request, or an encryption request or a request to make calculations on the encrypted data stored in the public cloud.

Our Private Core Cloud has several interconnected servers that are managed by a management server controlled by the administrator. So, the A-HEaaS Core Cloud has a key management server that generates encryption keys, an authentication server that allows users to authenticate to the cloud service provider and an encryption server using only crypto systems with homomorphic properties. The management server is responsible for monitoring our private core cloud, auditing and updating the used security policies.

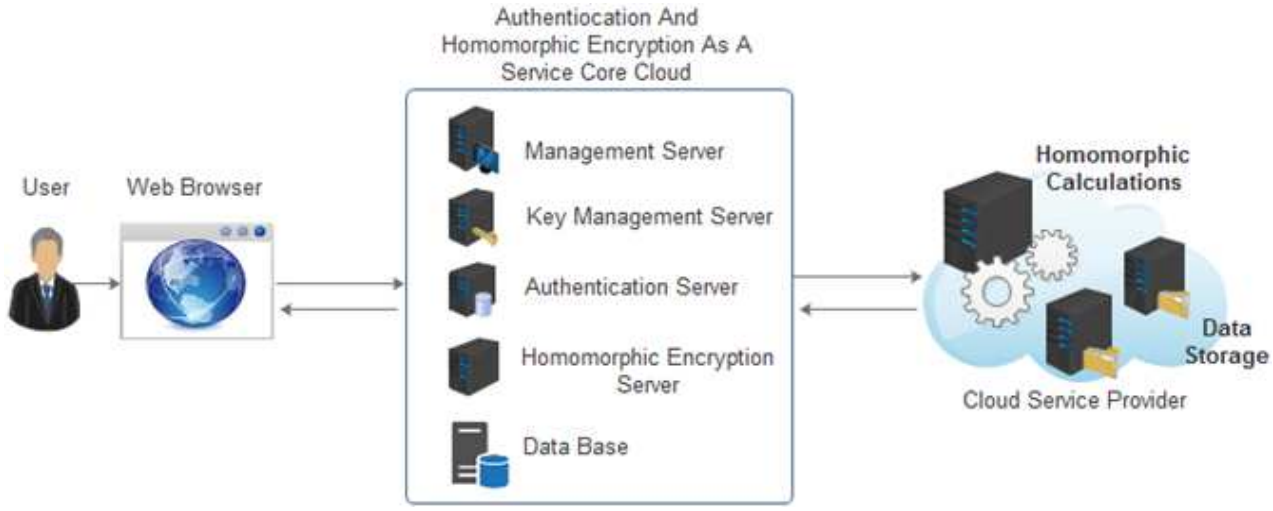

Fig. 1: Global view of A-HEaaS model

The homomorphic calculation server that makes calculation on the encrypted data will be deported to the cloud service providers. So, we will make the separation between the homomorphic calculations server and the key generation server and this architecture will also allow us to use the resources of the cloud to make calculations on the encrypted data without sharing the encrypted keys with the cloud service provider.

Our framework operates in three distinct phases which will be detailed in the subsections below: First we will use an authentication mechanism to secure access to the Cloud server. Secondly we will proceed to data encryption and thirdly we will proceed to the homomorphic computation.

Authentication as a Service

In this part, we will propose the Authentication System Access Manager (ASAM) based on Security as a Service model. This system will allow us to offer users a platform for authentication to the public cloud servers. Our system is hosted in a dedicated server set in a private cloud.

We will suppose that the users' protected data are inside a public cloud and the access is provided only after a successful login. The Authentication System Access Manager (ASAM) consists of a set of components designed to provide secure, authenticated communications between client applications and public Cloud services. The purpose of ASAM is to ensure that communication is both authenticated and secure. Figure 2 describes the architecture of ASAM.

The ASAM system is composed of:

- A Key Management Server which is responsible for generating the encryption and decryption keys used to design the authentication request
- An Access Management server which is responsible of user profiles management
- An Authentication Server which handles the authentication mechanisms

The ASAM service maintains a database of registered users credential and the end-users communicate with ASAM service using a web application with https transactions. Using the ASAM system, each user will have the ability to authenticate to remote public cloud servers securely. The workflow example; Fig. 3 describes the steps of authentication mechanism to the public cloud services.

In the first step, the user must register with ASAM system using the web application. To access to the authentication services offered by our ASAM system, the user must authenticate using his login and password created during registration. Based on the RFC 6238 (M'Raihi, 2011), we built a One-Time-Password (OTP) authentication with two factors. So, for each new session it's allocated a unique, randomly generated session token and will be discarded at the end-of-session. This is sent to the users' to add more safety, because security credentials alone are no longer enough for a secure access to the public cloud. The two-factor authentication enhances the level of security of the authentication process.

After successful login, the token will be generated by the authentication server and will be sent to the users' device. Once the user enters the token through the web page, the Authentication server generates an authentication request to ensure the connection to the public cloud.

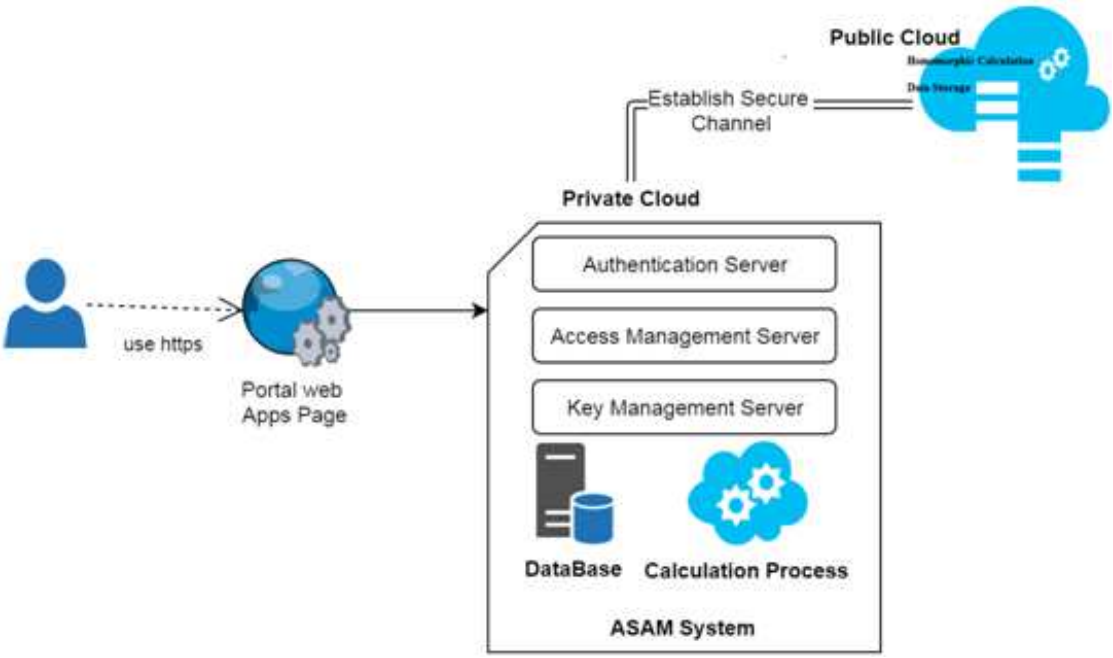

Fig. 2: General architecture of ASAM System

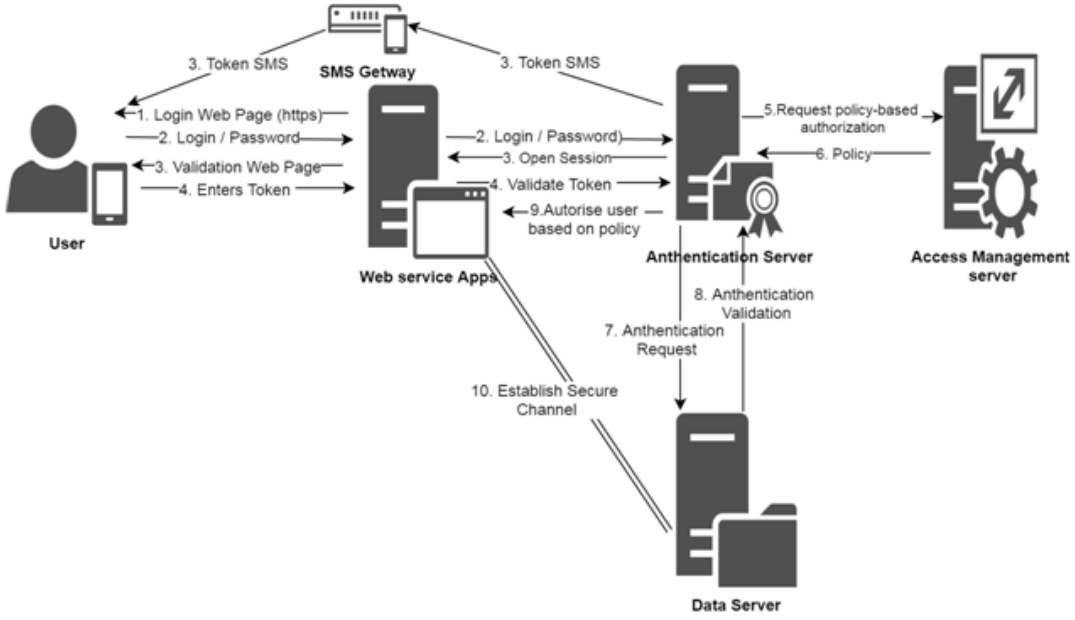

Fig. 3: Authentication as a Service mechanism

Each token is single-use and it associated to the User ID. The token is automatically destroyed after the end of session which enhances the level of security. The request will be generated after the beginning of each session and this request will be encrypted using a Symmetric-key algorithms.

On our model, we will use the AES crypto system because it uses a small key size with a good level of robustness which can help us to manage a big number of users at the same time. The authentication process is explained below:

1. The ASAM key Management server generates a new AES key k associated to the user ID (UID)
2. The ASAM key Management Server KMS sends the key to the ASAM Authentication Server AS
3. The ASAM authentication server AS encrypts the token and the user ID with AES crypto system using

the generated key k and hash the *User ID* (*UID*) and Token with SHA-3 then sends information to the Public Cloud (PC):

$$Auth_{Req} = E_k\left(UID \oplus Token\right); Hash\left(UID \oplus Token\right) \quad (3)$$

$$AS \rightarrow PC : E_k\left(UID \oplus Token\right); Hash\left(UID \oplus Token\right) \quad (4)$$

4. The Public cloud decrypts the message to get the user ID and the token (*UID* + *Token*). If the user exists, the public cloud associates the token and the key k to the user and open a new session

The hash information is used to generate a digital signature and then check the integrity of the authentication requests and responses messages.

In this part we have constructed an authentication as a service model using an authentication scheme based on the AES crypto system. We will then proceed to the construction of the second part of our model which will allow us to offer calculation and homomorphic encryption services to the concerned users.

Homomorphic Encryption as a Service

The main purpose of the second part of our framework is to use the resources of the Cloud to make calculations on encrypted data, using the homomorphic encryption. For this we must first encrypt data using homomorphic crypto system. In our model we will use a somewhat/fully homomorphic encryption system to encrypt data.

When implementing this model it is possible to use any other homomorphic crypto system because the purpose of this architecture is to demonstrate the possibility of using homomorphic encryption in a SECaaS framework and not the study of the crypto system itself. Regarding homomorphic crypto systems there are three distinct types:

1) The first type concerns the Multiplicative Homomorphic Encryptions: For example RSA is homomorphic for multiplication (Rivest *et al*., 1978), as shown below:

Let's choose tow cipher texts $E(M_1)$ and $E(M_2)$, the multiplication is made as:

$$\begin{aligned} &E(M1) \otimes E(M2) = (M1)^e (M2)^e \, mod n \\ &= (M1M2)^e \, mod n = E(M1 \otimes M2) \end{aligned} \quad (5)$$

2) The second type concerns the Additive Homomorphic Encryption: For example PAILLER is homomorphic for addition (Paillier, 1999), as shown below:

Let's choose tow cipher text $E(x_1; r_1)$ and $E(x_2; r_2)$. Let a random $g \in Z_n^*$ as $L = (g^{\lambda} \, mod n^2)$ is invertible where, $L(u) = \frac{u-1}{u}$. The addition is made as:

$$\begin{aligned} &E(x_1; r_1) \oplus E(x_2; r_2) = g^{x1+x2}(r_1 r_2) mod n^2 \\ &= E(x_1 \oplus x_2; r_1 r_2) \end{aligned} \quad (6)$$

3) The third type represented by somewhat/fully Homomorphic Encryption: We have chosen to use the DHGV which is homomorphic for addition and multiplication (Dijk *et al*., 2010), as shown below:

Let's choose tow cipher text $E(x_1)$ and $E(x_2)$ and random large r. The addition is made as:

$$E(x_1) \oplus E(x_2) mod p = 2(r_1 \oplus r_2) \oplus x_1 \oplus x_2 \quad (7)$$

Let's choose tow cipher text $E(x_1)$ and $E(x_2)$, random large r. The multiplication is made as:

$$E(x_1) \otimes E(x_2) mod p = 2(2 r_1 r_2) + r_1 x_1 + r_2 x_2) + x_1 x_2 \quad (8)$$

Using a somewhat/fully homomorphic encryption system, we will build an encryption and decryption as a service model "ENC/DECaaS" that will encrypt users' data through the use of a private cloud server "Encryption Server". Figure 4 illustrates the main components of our ENC/DECaaS model.

The proposed model ensures the confidentiality of data by using a web application server, a key management server to generate the encryption keys and a private cloud server "Encryption Server" to encrypt and decrypt data. The encrypted data will be stored in a remote public cloud server. The encryption and decryption as a Service model process is explained below.

Authentication Process

In order to access the different security services of our A-HEaaS model, each user must authenticate to the web application made available to him. The user will then be able to download or upload files from or to the remote public cloud servers. The web server verifies the identity of each user by sending an authentication request to the authentication server and by using our authentication as a service model.

Key Generation Process

To encrypt and decrypt data, it is essential to generate appropriate encryption keys. Those encryption keys will be generated using a key management services.

To do so, we will use varying sizes of keys, which vary according to the level of sensitivity of data and to desired action (authentication, encryption or calculations) and we will add a key lifecycle management policy to protect against loss of encryption keys. The key generation procedure also varies depending on the choice of the crypto system to be used.

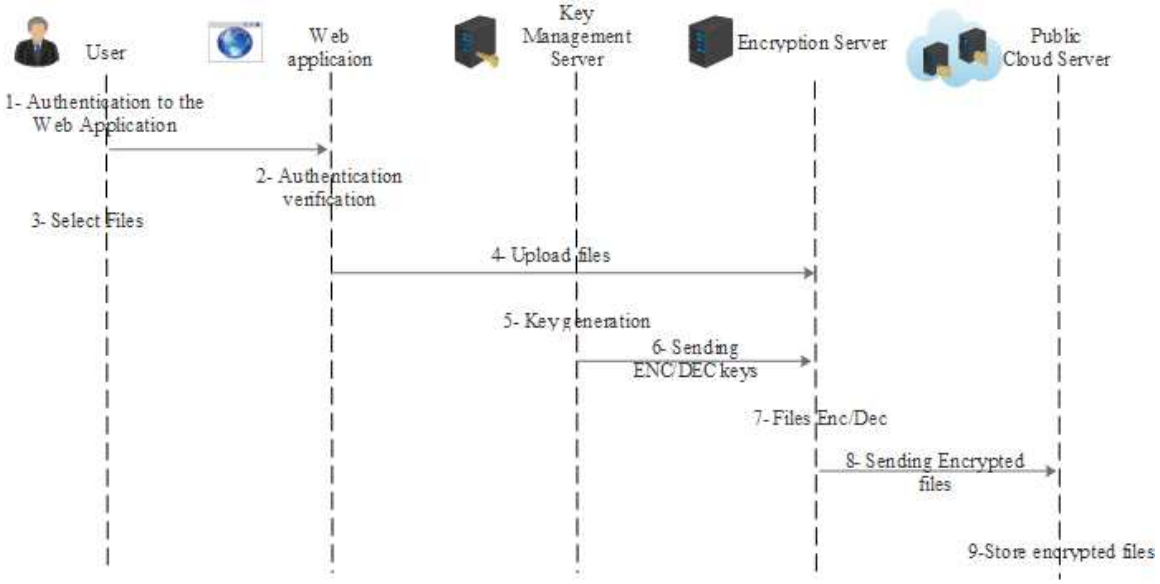

Fig. 4: Encryption as a service model process

In our case we will use the encryption system DHGV. So, the key generation is done as follows:

Let p and q two large prime and a random large r as $r \sim 2^n$, $p \sim 2^{2n}$ and $q \sim 2^{5n}$.

So, p is the private key and $E(x) = pq + 2r + x$. X represent one bit of the information.

The key management server will be located in our Private Core Cloud, so access to this server will not be possible except for authorized people and the sessions will be automatically closed after the end of each operation.

Our key management server is separated from the cloud provider which hosting users' data. This separation enhance the security level of our architecture and give a protection against external breaches as well as an attack originating from a privileged of the Cloud provider.

Encryption Process

The encryption server will then encrypt or decrypt the data: The encryption server will request the service of the key management server to retrieve the encryption keys. After that he will be responsible for encrypting data, using only FEH crypto systems. The encryption process is done as follows.

Encryption

The cipher text is calculated as:

$$E(x) = pq \oplus 2r \oplus x \tag{9}$$

Decryption

To retrieve the message we calculated as:

$$D(y) = (ymodp) mod 2 \tag{10}$$

Then a signature will be generated to guarantee the data protection before sending the encrypted data to the remote public cloud servers.

Homomorphic Calculations as a Service

When a user uses the services of a cloud provider, it risks losing the privacy and confidentiality, especially if he often uses the cloud resources to do some calculations on his data. Precisely, when a user wants to make calculations on stored data in the Cloud, the server will ask the user to provide him the encryption key in order to decrypt the data and then it will make the various calculations then re-encrypt the data and the obtained results, before sending them to the concerned user. In this procedure, it is clear that the administrators of the cloud services may view the user's personal data in clear which can cause loss of confidentiality and privacy.

The Homomorphic encryption offers the possibility of doing calculations on encrypted data without decrypting them. So, the Homomorphic encryption may rectify this problem and save privacy and confidentiality issues, but unfortunately it is not yet usable except in some few specific cases because it still suffers from several limitations:

- Firstly the size of the key required to make the calculations must be very large
- Secondly the time required to make these calculations is enormous and it increases exponentially by increasing the size of the data that we want to calculate
- Finally there are no cryptosystems that can be qualified as completely homomorphic, so we must

add additional calculations if we want to do addition and multiplication at the same time

Table 1 shows some standards homomorphic calculation results done in traditional environment, using several keys and text sizes: We note A the addition of two encrypted messages C_1 and C_2 such that $A = C_1 \oplus C_2$ and we denote M the multiplication of two encoded messages C_1 and C_2 such that $M = C_1 \otimes C_2$.

It is clear that the time of calculation is very long. So, to address all these limitations, we propose a model of homomorphic calculations as a service "HCaaS" that allows to make calculations on the encrypted data by using cloud's resources. To do so, we developed a dedicated server that makes calculations on encrypted data. This procedure allows us to get benefit from the infinite resources of the Cloud and to reduce significantly the time required for calculations.

If we want to use the Cloud resources in order to make the homomorphic calculation, we must be able to manage the Encryption/Decryption keys and keep them confidential.

So the idea of our model (HE-aaS) is to keep the encryption keys away from the cloud providers and use a dedicated private core cloud to manage those keys and the calculation requests issued by users. To do so, we will use our private core cloud and more specifically we will use our key management server. Figure 5 shows the different components of our model.

It is assumed first that the user's data is stored in the remote public cloud servers, the keys generation is done by the key management server and the encryption and decryption of the data is done by the ENC/DEC server. To make calculations on encrypted data, the procedure is as follow:

(1) Firstly, the user generates and sends a request of calculations to the Private Core Cloud
(2) Secondly the request will be redirected to the remote Cloud Servers
(3) After receiving the calculation request, the cloud provider will starts the homomorphic calculation procedure. This calculation will be making in blind because the cloud will make calculation on encrypted data and without having a decryption key
(4) Finally, the result of these calculations will be sent to the ENC/DEC server located on our private core cloud to be decrypted. So, the user can retrieve his data and the result of his calculation request in clear

Table 1: Homomorphic calculation

Homomorphic encryption		Calculation/key size (ms)							
		Key size: 128 K				Key size: 256 K			
		A	D(A)	S	D(S)	A	D(A)	M	D(S)
Text size	100 Ko	56589	30658	63245	45687	252487	147856	301253	265203
	10 Ko	142354	128349	154893	107631	756948	546897	771204	548723

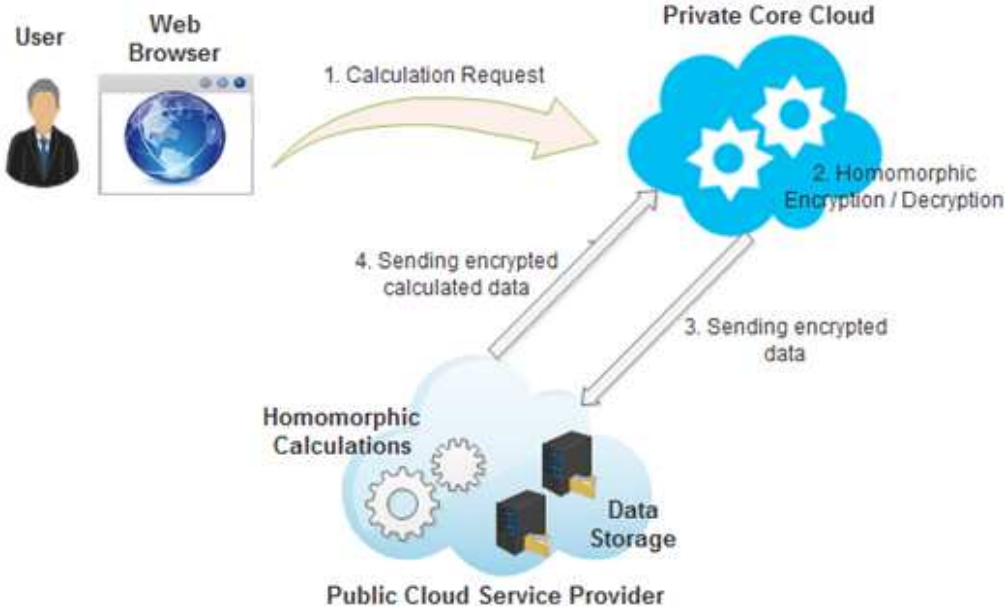

Fig. 5: Homomorphic calculation as a service mechanism

A-HEaaS Implementation

Implementation of A-HEaaS Model

In this part we will make an implementation of our A-HEaaS model using some medical data. We have chosen this example because it will allow us to make calculations using addition and multiplication.

We have designed a web application, which allows to enter the personal information of patients and to specify their diseases and their nationalities. These data will be encrypted by homomorphic crypto systems on our Private Core Cloud and they will be sent to remote Cloud servers for storage.

The purpose of this application is to make calculations on these data and to obtain some statistics related to these patients according to the type of their disease and to their nationality.

First, the user has to register in our web application and send an authentication request to our authentication server. Table 2 shows an example of the different steps results of the authentication mechanism.

After the authentication step, the users will have the possibility to access to the web platform linked to our model. The connection between the server and the web application is made by the HTTPS protocol.

The web interface allows consulting the database adding new entries and to looking for patients by their type of diseases or their nationalities. Table 3 shows the results of the steps for data encryption and homomorphic calculations.

The data entered by the users will be processed and encrypted by our Private Core Cloud, by soliciting the key management server and the homomorphic encryption server.

We will then make calculation on these data and proceed to data processing using homomorphic calculation mechanism. We will send a request to the server to know the number of patients of Moroccan nationality and who suffer from the disease A. So the Encryption Server will encrypt the request and send it to the Cloud. The Cloud will search the corresponding data on its database and then he will make the calculation on these encrypted data.

The processing will be done on the encrypted data and the result will be decrypted by our homomorphic encryption server before being sent to the concerned user.

Evaluation and Security Analysis

In this section we will evaluate our A-HEaaS model compared to existing frameworks based on SECaaS model. The purpose of our evaluation is to show that our design satisfies the cloud users' expectations in terms of confidentiality, authenticity, integrity and privacy. Table 4 shows the efficiently and the robustness of our framework compared to the existing model.

We will proceed in the following to a security analysis which will allow us to evaluate our A-HEaaS model. We will prove that our model can tackle all these securities issues very efficiently and offer to cloud users and companies the possibility to manage their security using cloud services while preserving their privacy.

Concerning authentication and identity access management, we have developed a system based on several authentication parameters to increase the robustness of the access verification mechanism. The schema used is based on a SECaaS model because it uses a processing authentication queries servers located in the cloud. The different queries are encrypted using the AES system crypto. Encryption keys are automatically regenerated after the end of each session.

Table 2: Authentication as a service process

Authentication process	
Login	Karim.zkik
Password	7Xx9C2Az
Password Hash (sha3-512)	1c4b655fd6794a5ee764e793c05a37bbc5fbfa2ca6629114 dd87c14bae6a11398c276c77e7700f16bd86e555142b3f02 523cbe 0364f7b7898f0abdfd3e02abc
ID	215148037
Token	459873
AES key size	128/256
Encrypted Authentication Request (AES)	Q3WXEsd+9vBf5NFeoV2dbw==
Authentication Request Hash (sha3-512)	7be08ca42a67eec897f1f5bda6dcb87426c7a9660dcbe0 1f8c485b330d6c7bb633f9e9cd03a3ce66abbae03581a02 b49e338eb353ed927be5f0a6d1f48a1a4b2
Signature	WXlu+tWi45r9+wszdOvUEictiWQGRIZhxAY3M2QFMV +cPD+lP3KdC33C8qrfjjHRcm4B067JjmS+nra4W1uJ9Cu 5/zjh1Ys96tad6ZDXgvq6xperRC2Lm4eY+MWRAq11lFlz 3WvfaSMoCEzXQxjwyAhOgyyErXwivQs0n8woiu1aBIw Mciml2XqEzSAarmM
Encryptions time	1,82614 ms

Table 3: HEaaS model precess

Encryption setting	
Key size	1024
Block size	4
Private key	P=13432170912101400848…156985366996877090730111 96
	O=14857479576017431699…41273676993571540617492 3483
	D=14867067979894749080…79140490325401937739495 4033
Public key	**N**=19956820498812199991…65114654984598936319808 0163
	F=57363120267286726563…28635118767116488484762 5457
Homomorphic encryption process	
Data	(M1, M2)=(A, Disease Morocco)
Encrypted data	**E(M1)**=07068090566679436352294337428641883767152997 44800538373637182468939167573072563278030708580098 3755662216927445766080277973973165793876557475632464 15085323837871420278159706829021746568069874949716 0449968834604570667428576063467437647417782280179193 45680622…32254599664845784085987755655895
	E(M2)=04845778776207359567039126457593993334683940 15998279843105242158489484286224643303695678774478 1520638320601348842391873837941870754048703572036190 16330914744142827086140844121750397147983931477917 1913565444889385382616342442437552208279016386344785 11571654….23883774049063461669908631379452
Encryption time	572 ms
Homomorphic Calculation	
Request	The number of patients who have disease A and who are Moroccan
EncR= encryptedresults	10162018256734010175171449822771603263094434490911 4752911048966048641576766949379676486305672514242746 0659383190298833874487381457481989 5…36294310495666 621348821828075856
Dec(EncR) = numbero fpatients	156
Calculation time	4257 ms

Table 4: Comparison of security frameworks based on secaas model

	Gupta and Gedam (2014)	Chen *et al*. (2014)	Rajkumar (2014)	Elsayed and Zulkernine (2016)	Sharma *et al* (2016a; 2016b)	Wüller *et al*. (2017)	Hayward and Chiang (2015)	Proposed model
Authentication and identity management	X				X			X
Integrity		X		X				X
Confidentiality	X		X	X	X			X
Privacy		X		X		X	X	X
Key management					X	X		X
Intrusion prevention		X	X					X
Homomorphic encryption						X	X	X
Perfomance management			X	X			X	X

The system used is based on the one time password concept to reduce the possibilities of encryption key and passwords leakage. User redirection to different cloud servers for data storage is made via secure channels using VPN tunnels and secure protocols such as TLS / SSL. The security policy used for the authentication process is periodically updated and monitored by the access server management.

To guarantee the security of the Cloud users, we have proposed an encryption mechanism based on homomorphic cryptosystems. The choice of this type of crypto system is very costly in terms of performance, but it will give a high level of security and it will allow us to make calculations on encrypted data without decrypting them.

On the other side, the use of the homomorphic computation allowed us to preserve the privacy of the users. The use of a SECaaS model allowed us to fully benefit from the resources of the Cloud which gave us the possibility to circumvent the limitations of homomorphic computation. We used simple calculations on small data to prove the possibility of using homomorphic encryption in a Cloud environment.

Parallel processing of queries and calculations can further reduce the computing time and increase performance.

We used several crypto homomorphic systems during our implementation but there are still several crypto system types FEH that can be used which can increase the power and the speed of calculations.

Several companies do not use cloud services because of the different threats related to the cloud services providers. Even if some Cloud providers, provides to their customers with the protection of their data stored in cloud servers, it is better not to completely trust their data protection system. The use of homomorphic encryption and the use of an encryption key management mechanism allow us to bypass this problem so that we can make full use of the resources of the Cloud. Thus, the data storage, processing and calculation will be done in blind.

In the following, we will analyze our authentication mechanism, Fig. 6, by comparing the time required for each authentication step according to the used key size.

Figure 7 shows the time required to generate encryption keys, to encrypt data and to make calculation on this encrypted data. The encryption and calculation process are made using different key sizes and on 100Mo data size.

Performance management is a very important aspect in the development of our A-HEaaS model. To demonstrate the robustness of this model, we have made a comparison of CPU consumptions on our servers during the different processes of our model according to different key sizes. Figure 8, shows the results of this comparison and we can conclude that our framework not overload server while using 1024 or 2048 key sizes.

Our access management server handles packets passing between the remote cloud servers and the end users. This server allows, among other things, to update the security policy and to analyze the packets according to some number of security rules. This server makes it possible to secure the network and to avoid flow attacks (for example DoS and DDoS attack). It can also avoid intrusions and to stop abnormal traffic. Figure 9, shows the result of the analysis of several traffic with different rates and shows the time required for the traffic analysis, the detection of anomalies and intrusion, the resolution of violations and the time necessary for the rejection of malicious traffic.

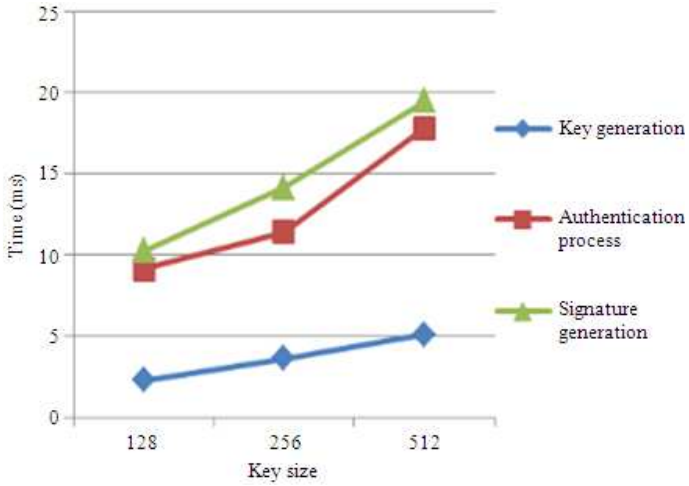

Fig. 6: Authentication process

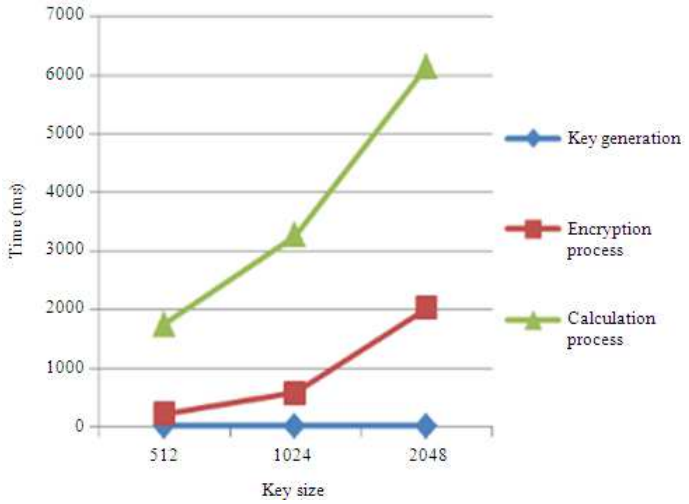

Fig. 7: Encryption and calculation process

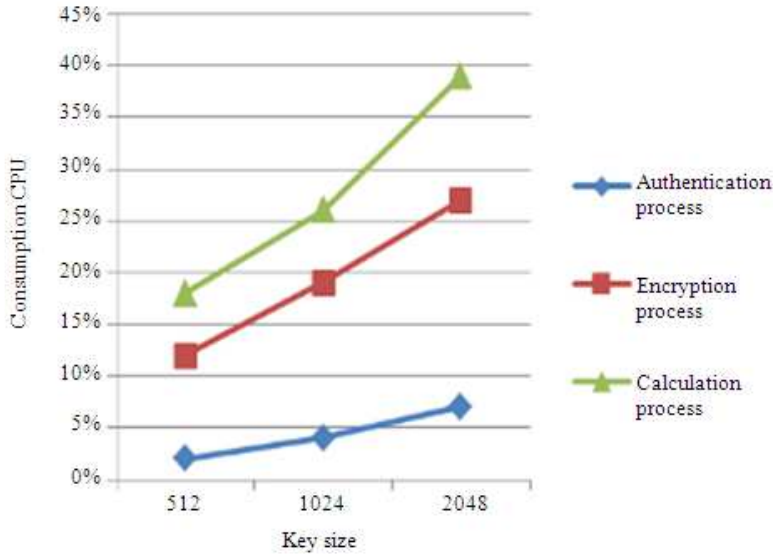

Fig. 8: Performance analysis

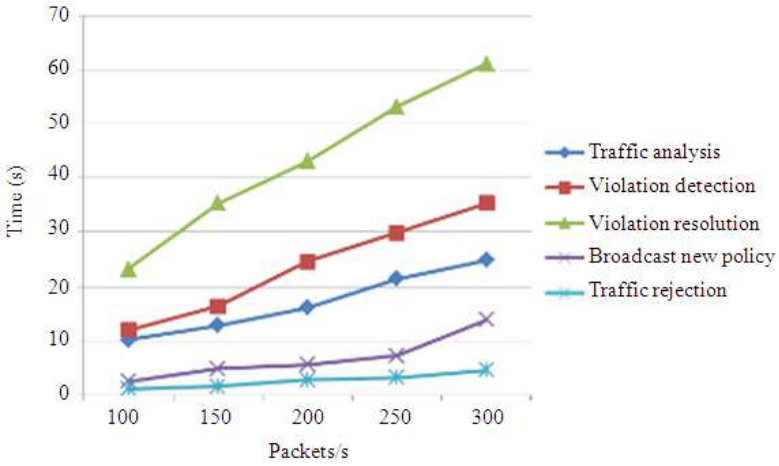

Fig. 9: CPU consumption

Intrusion and attacks prevention is a very important aspect regarding the security of user data. To do this, we plan to add an IPS in inline mode that will allow us to filter the traffic and to detect and stop the various intrusions. Our IPS server must be linked to the Cloud so it can download the latest signatures of malwares and protect Cloud users against any potential attack.

Our evaluation shows that our design satisfies the security requirements and that allows users to authenticate, store and make calculation on their data safely. Our scheme uses homomorphic encryption schemes which ensures confidentiality and privacy and keeps a high level of security. The use of various resources of the cloud has enabled us to establish a pattern of use of homomorphic encryption, which ensures a good performance management and speed of calculations appropriate to users' expectations.

Conclusion

Security as a service is a new model that offers huge benefits for users and companies in terms of security. The purpose of our work is to build a secure architecture for authentication and homomorphic encryption based on Security as a service model. So we develop a model that can allow user to make a secure access to the cloud service providers using an authentication as a service model and allow him to encrypt his personal data and make treatments on theirs encrypted data using an homomorphic encryption as a service model.

We develop a web application to allows user access to our Private Core Cloud and we make an implementation of our model using a medical application. So each authorized user can consult the patients' data base and make some request to know some

confidential specifics information using a homomorphic calculation. On our model we benefits of Cloud's resources to make faster calculation and response to customer requirements in terms of efficiency and safety. Regarding our future work we would like to develop a model that could permit secure data sharing, integrates this A-HEaaS model with various other SECaaS services and it is expected to develop more suitable encryption model using parallel processing and other homomorphic encryption schemes to enhance performance and security level.

Acknowledgement

This work has been achieved at Laboratory of Mathematics, Computing and Applications at Faculty of Sciences, Mohammed V University in Rabat.

The author would like to thank everyone who has contributed to the progress of this research.

Author's Contributions

Karim Zkik: Establishment of the state of art, Contribute to the development of the three parts of the A-HEaaS model, drafting and writing the article (about 50-60%), the acquisition, analysis and interpretation of data.

Maha Tebaa: Contribute to the development of the Homomorphic encryption and calculations as a service model. Contribute in writing the article (about 15-20%), the acquisition, analysis and interpretation of data.

Tarik Tachihante: Contribute to the development of the Authentication as a service model. Contribute in writing the article (about 10-20%), the acquisition, analysis and interpretation of data.

Ghizlane Orhanou: Contribute to the development of the three parts of the A-HEaaS model. Contribute in drafting and writing the article (about 10-20%), reviewing the paper, analysis and interpretation of data.

Ethics

This article is the original contribution of the authors and is not published elsewhere. There is no ethical issue involved in this article.

References

Alliance, C.S. and S. Secaa, 2011. Defined Categories of Service. 1st Edn., Cloud Security Alliance, pp: 27.

Alliance, C.S. and S. Secaa, 2012a. Implementation Guidance: Identity and Access Management. 1st Edn., Cloud Security Alliance, London, UK.

Alliance, C.S. and S. Secaa, 2012b. Implementation Guidance: Encryption. 1st Edn., Cloud Security Alliance, London, UK.

Amna, A., 2013. A comparative study of fully homomorphic encryption schemes for cloud computing. Int. J. Emerging Technol. Adv. Eng., 3: 50-54.

Chen, Y.C., Y.S. Wu and W.G. Tzeng, 2014. Preserving user query privacy in cloud-based security services. J. Comput. Security, 22: 997-1024. DOI: 10.3233/JCS-140520

Delamore, B. and R.K. Ko, 2015. Security as a Service (SecaaS)—An Overview. In: The Cloud Security Ecosystem: Technical, Legal, Business and Management Issues, Katsaropoulos, C. (Ed.), Elsevier - Syngress, pp: 187-203.

Dijk, M.V., C. Gentry, S. Halevi and V. Vaikuntanathan, 2010. Fully homomorphic encryption over the integers. Proceedings of the Annual International Conference on Theory and Applications of Cryptographic Techniques, May 30-Jun. 03, Springer-Verlag Berlin, French Riviera, France, pp: 24-43. DOI: 10.1007/978-3-642-13190-5_2

Elsayed, M. and M. Zulkernine, 2016. IFCaaS: Information flow control as a service for cloud security. Proceedings of the 11th International Conference on Availability, Reliability and Security, Aug. 31-Sept 2, IEEE Xplore Press, Salzburg, Austria, pp: 211-216. DOI: 10.1109/ARES.2016.27

Furfaro, A., A. Garro and A. Tundis, 2014. Towards Security as a Service (SecaaS): On the modeling of security services for cloud computing. Proceedings of the International Carnahan Conference on Security Technology, Oct. 13-16, IEEE Xplore Press, Rome, Italy, pp: 1-6. DOI: 10.1109/CCST.2014.6986995

Gentry, C., 2009a. A fully homomorphic encryption scheme. PhD thesis, Stanford University, USA.

Gentry, C., 2009b. Fully homomorphic encryption using ideal lattices. Proceedings of the 41st Annual Symposium on Theory of Computing, May 31-Jun. 02, ACM, New York, USA, pp: 169-178. DOI: 10.1145/1536414.1536440

Getov, V., 2012. Security as a service in smart clouds—opportunities and concerns. Proceedings of the IEEE 36th Annual Computer Software and Applications Conference, Jul. 16-20, IEEE Xplore Press, Izmir, Turkey, pp: 373-379. DOI: 10.1109/COMPSAC.2012.112

Gupta, S. and N.R. Gedam, 2014. Security service model for cloud environment. Int. J. Sci. Res., 3: 1359-1361.

Hayward, R. and C.C. Chiang, 2015. Parallelizing fully homomorphic encryption for a cloud environment. J. Applied Res. Technol., 13: 245-252.

Hemalatha, S. and R. Manickachezian, 2014. Performance of ring based fully homomorphic encryption for securing data in cloud computing. Int. J. Adv. Res. Comput. Commun. Eng., 3: 8496-8500.

Khan, M.A., 2016. A survey of security issues for cloud computing. J. Netw. Comput. Applic., 71: 11-29. DOI: 10.1016/j.jnca.2016.05.010

M'Raihi, D., 2011. RFC 6238 - TOTP: Time-Based One-Time Password Algorithm. 1st Edn., Internet Engineering Task Force (IETF), United States.

Marketsandmarkets.com, 2016. Security as a service in smart clouds—opportunities and concerns, security as a service market by solution (Email encryption, SIEM, IAM, endpoint protection, IDS/IPS, data loss prevention and others), by service, by application, by organization size, by vertical, by region - global forecast to 2020.

Paillier, P., 1999. Public-key cryptosystems based on composite degree residuosity classes. Proceedings of the 8th Annual EUROCRYPT Conference (EUROCRYPT'99), Prague, Czech Republic 1592, pp: 223-238.

Poteya, M.M., C.A. Dhote and D.H. Sharma, 2016. Homomorphic encryption for security of cloud data. Proceedings of 7th International Conference on Communication, Computing and Virtualization, (CCV'16), pp: 175-181. DOI: 10.1016/j.procs.2016.03.023

Rahmani, H., E. Sundararajan, M.A. Zulkarnain and A.M. Zin, 2013. Encryption as a Service (EaaS) as a solution for cryptography in cloud. Proced. Technol., 11: 1202-1210. DOI: 10.1016/j.protcy.2013.12.314

Rajkumar, M.N., 2014. Providing flexible security as a service model for cloud infrastructure. Int. J. Adv. Res. Comput. Eng. Technol., pp: 3924-3930.

Rivest, R., 2002. Lecture notes 15: Voting, homomorphic encryption. Comput. Netw. Security, 4: 17-31.

Rivest, R., A. Shamir and L. Adleman, 1978. A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM, 21: 120-126. DOI: 10.1145/359340.359342

Samlinson, E. and M. Usha, 2013. User-centric trust based identity as a service for federated cloud environment. Proceedings of the 4th International Conference on Computing, Communications and Networking Technologies, Jul. 4-6, IEEE Xplore Press, Tiruchengode, India, pp: 1-5. DOI: 10.1109/ICCCNT.2013.6726636

Sharma, H.D., C.A. Dhote and M.M. Potey, 2016a. Identity and access management as security-as-a-service from clouds. Proceedings of the 7th International Conference on Communication, Computing and Virtualization, (CCV'16), pp: 170-174. DOI: 10.1016/j.procs.2016.03.117

Sharma, H.D., C.A. Dhote and M.M. Potey, 2016b. Intelligent transparent encryption-decryption as security-as-a-service from clouds. Proceedings of the International Conference on Computational Systems and Information Systems for Sustainable Solutions, Oct. 6-8, IEEE Xplore Press, Bangalore, India, pp: 359-326. DOI: 10.1109/CSITSS.2016.7779386

Subashini, S. and V. Kavitha, 2010. A survey on security issues in service delivery models of cloud computing. J. Netw. Comput. Applic., 34: pp: 1-11. DOI: 10.1016/j.jnca.2010.07.006

Tebaa, M., K. Zkik and S. El Hajji, 2015. Hybrid homomorphic encryption method for protecting the privacy of banking data in the cloud. Int. J. Security Its Applic., 9: 61-70. DOI: 10.14257/ijsia.2015.9.6.07

Varadharajan, V., 2014. Security as a service model for cloud environment. IEEE Trans. Netw. Service Manage., 11: pp: 60-75. DOI: 10.1109/TNSM.2014.041614.120394

Werner, J., C.M. Westphall and C.B. Westphall, 2017. Cloud identity management: A survey on privacy strategies. Comput. Netw., 122: 29-42. DOI: 10.1016/J.COMNET.2017.04.030

Wüller, S., D. Mayer, F. Förg, S. Schüppen, and B. Assadsolimani *et al*., 2017. Designing privacy-preserving interval operations based on homomorphic encryption and secret sharing techniques. J. Comput. Security, 25: 1: 59-81. DOI: 10.3233/JCS-16830

Zkik, K., G. Orhanou and S.E. Hajji, 2017. Secure mobile multi cloud architecture for authentication and data storage. Int. J. Cloud Applic. Comput., 7: 62-76. DOI: 10.4018/IJCAC.2017040105