Data Security in healthcare Systems

Igor Godefroy Kouam Kamdem, Marcellin Julius Antonio Nkenlifack

Data Security in healthcare Systems

Igor KOUAM KAMDEM[*1] **and Marcellin NKENLIFACK**[2]

[1,2]URIFIA Laboratory, University of Dschang, Cameroun

*E-mail : igorkouam5@gmail.com

Abstract

Computer security is a concept that aims to protect computer systems against all violations, intrusions, damage or data theft within an information system. This article focuses on data security while guaranteeing security services such as confidentiality, integrity, authenticity, availability and non-repudiation. Our main objective is to propose a solution that can protect data in information system. The model proposed in this paper ensures security at two levels: at the local level (protection of the server) with the data's distribution in several classes to ensure real-time availability; and at the network level during the routing or exchange information. In the context of exchanges, the model integrates a biometric authentication to validate the operation. We have also introduce multi-stage deception technique. This makes possible to guarantee authenticity and integrity simultaneously. The integration of deception technique allow us to ensure robust security. In addition, the use of encryption algorithms ensures confidentiality during exchanges. At each level, we also distinguish between hardware protection and software protection.

Keywords

Cybersecurity, E-Health, Information system, Security, Cryptography

I INTRODUCTION

Security is a concept that nowadays extends to all fields, i.e. automotive, aviation, medicine, education and others. The digitization of systems has considerably changed things because we observe a gain in time, money, resources etc. In the case of health, for example, we talk about E-health, which is the use of IT tools in the health field. Over the years, the transformation of the health-care sector through digital technology has become a top priority for health-care institutions. This transformation responds to two fundamental challenges: improving organization by facilitating the work of professionals, and meeting the needs of users and residents [15].

The digitization of systems has given rise to the problem of security, which today is causing very serious damage. We have for example the data's theft as observed in the attack perpetrated in France which exposed on the Dark web on February 25, 2021 where the data of nearly 500,000 French patients [11] have been exposed; the paralysis of the computer system, with the attacks perpetrated at the hospital center Dax on February 09, 2021 [14] and that of Ville Franche on February 15, 2021 [13], which completely paralyzed the system of these health centers.

This article focuses on data security at the local level as well as at the network level during a transfer while introducing cyber security. The objective is to propose a security model that can address the security issues encountered in real life. The proposed model is a local security

architecture that partially encrypts data. As for the security of the transactions, a strong authentication is required to authorize the shipments; this allows to limit the fraudulent shipments in the system while detecting the intruders. Our contribution is twofold:

1. To propose a security model based on data classification for data security in a local server;
2. Exchange security with biometrics to control data exchange and prevent fraudulent sending.

The rest of paper is organized as follows: in section 2, we present a brief state of the art on previous work on data security; section 3 present our security model and detail each step of the model; section 4 show results and discussion, and we end with a conclusion in section 5.

It should be noted that this work is an extended version of a paper presented during the "Computing Conference 2021" and published in "Intelligent Computing, Proceedings of the 2021 Computing Conference, Volume 3" by Springer [**kouam2021**].

II STATE OF THE ART

According to the World Health Organization (WHO), "Health is a state of physical, mental and social wellbeing and not merely the absence of disease or infirmity" [**safon2029**]. Medical informatics has two main areas: medical practice with diagnostic support, computerization of the electronic patient record, economic optimization and others; but also research in medical science, which takes into account the following three criteria: internal validity, external validity and clinical relevance [10].

To ensure good security, it is necessary to define a security perimeter. "The security perimeter within the physical universe delimits the interior and exterior, but its definition must also encompass (or not) the immaterial entities that populate computers and networks, essentially software and in particular operating systems" [7].

Indeed, the legal aspect of health information is situated at three levels [1] : On personal data, which is defined by the French National Commission for Information Technology and Civil Liberties (CNIL) as "any information relating to a natural person likely to be identified, directly or indirectly"; Sensitive data, which are those that reveal, directly or indirectly, the racial or ethnic origins, political, philosophical or religious opinions or trade union membership of individuals, or relate to the health or sex life of these individuals, as defined by the CNIL; and the processing of personal data, which is the set of operations relating to these data.

Many works have been carried out on the protection of private data. Most of the works such as [3, 8, 9] propose a security for data storage in the Cloud. In 2011, Akinyele et al [3] developed a medical record management application compatible with Apple smart-phones. The data was encrypted and stored directly in the cloud. In 2014, Basel Katt [8] proposed a scheme based on encrypting the data before storing it in the cloud. The disadvantage of this scheme is the impossibility of searching the data, given that even the server on which it is stored is not able to decrypt it. Sun et al [9] provide a survey of different security techniques for private data stored in the cloud. This study reviews different security techniques and challenges related to software and hardware aspects to protect data in the cloud and aims to improve data security and privacy in a trusted cloud environment. A hybrid technique using both key sharing and authentication techniques is proposed by Rao [4] for data confidentiality and integrity. Mohammed at Al [5] proposes a three-layer security technique in the cloud namely the first layer for authentication, the second layer for data encryption and the third layer for data decryption. An event-based

approach to isolating critical data in the cloud is proposed [6], Trust Draw, a transparent security extension for the cloud that combines virtual machine introspection (VMI) and trusted computing (TC).

Katsikas and Al [2] have proposed a security architecture based on a trusted authority. This trusted authority also allows to certify the interactions between the two participants and guarantee that each one is who he claims to be. Mohammed MIROUD [10] developed in 2016 an access control model for health-care systems using Google's smartphone application, "Google authenticator". The particularity of this work is the use of one-time passwords. In addition, to optimize local security, the latter divides health data into three classes according to their sensitivity (non-sensitive, not very sensitive and very sensitive), as recognized by the health system.

The proposed cloud security solution suffers from one problem: data access. Indeed, Katt [8] specifies that it is complicated to perform a cloud search when these data are encrypted. It is therefore necessary to first decrypt the information, perform the operation and encrypt the data again before storing it. A second problem is related to the access time to the data. Mohammed Miroud [10] proposes for this purpose a solution of distribution and partial encryption of the data. Our model exploits the model proposed by [10] to address local security.

III DATA SECURITY MODEL

The proposed model aims to guarantee security services such as confidentiality, integrity, availability, authenticity and prevents data theft against any external person trying to break into the system. The model proposes security at two levels: at the storage server level and at the network level during data exchanges. We will present in detail the security at the local level first, and then at the network level.

3.1 Local security

Local security consists of protecting the data server against potential attacks.There are many problem at this level: attackers can consult and copy personals' data as we have seen with the publication of the data of French citizens [11]. At the same time, he could modify, create or delete the patients' files. Or he could block the system with a "ransomware" attack as in the attacks [14] and [13]. Local solution consists to solve these problems.

To solve this problem, we can encrypt server. But, this solution is very heavy to use because to access a data, we need to decrypt server (all data in the server) and encrypt it after consulting. Its takes too much time to do it and so, do not guaranteed data's availability. A good solution consist, when we want to consult data, to decrypt only data that we need. That is possible with partial encryption that consist to encrypt each data separately. Indeed, Mohammed Miroud [10] proposed a solution with partial encryption. We have exploited it solution to achieve our solution.

In [10], local data security consists in partitioning the data into three sensitivity classes: non-sensitive data, low-sensitive data and high-sensitive data. Using Rijndael's AES algorithm, he assigns a key size to each sensitivity class: a 128-bit key for non-sensitive data, 192-bit key for little-sensitive data, and a 256-bit key for highly sensitive data. Since the time to encrypt and decrypt data depends on the length of the key and the volume of data, he has thus reduced the time considerably with affordable security as well. The presented diagram in figure 1 is a schematic inspired by [10].

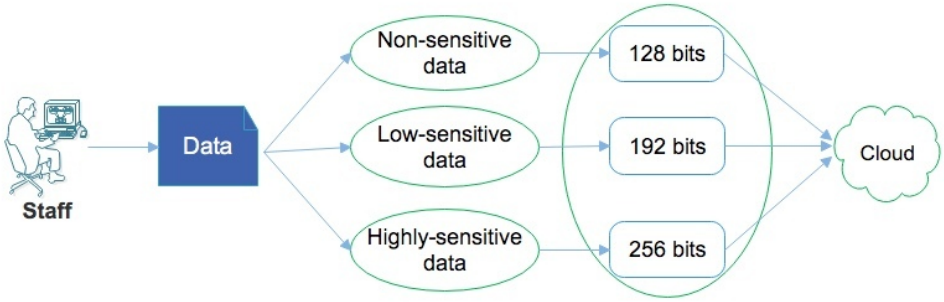

Figure 1: Data categorization inspired by Mohammed's model

Figure 1 show that each class of data is encrypted globally before being stored. Although the access time to the information is reduced compared to server's encryption, but it can also be not negligible with a large volume of data of each class.

Based on this model, rather than encrypting all the data's classes, our solution consists in encrypting each element of each category (each medical file in our case) independently before storage in the server as show at figure 2. Moreover, if in this model we consider non-sensitive data as those representing elementary information that does not allow to deduce anything about the patient's health status, it is possible to further reduce the access time to the information. To do this, the non-sensitive data will be stored in the system in clear as it is the primary data that is most consulted to identify patients. Ensuring spontaneous access to this data will make the system more efficient.

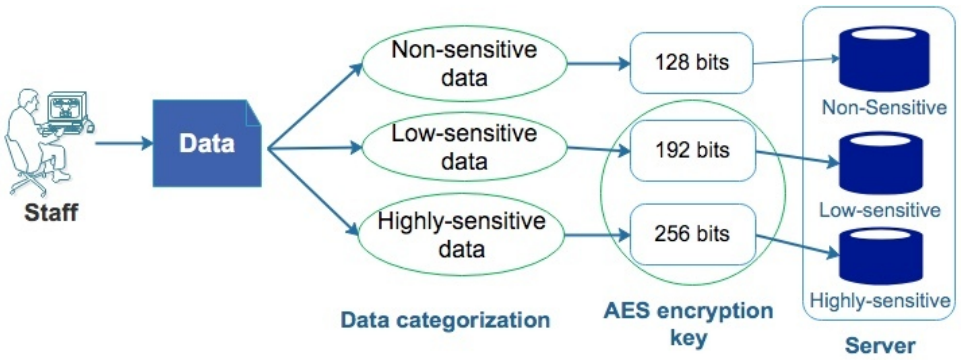

Figure 2: Local Security Model

Local security express by figure 2 can be describe by algorithm III.1 with the $DataCategorisation$ function.

The function takes as parameter the file to save and also the number of sensitivity classes to consider. In our case, we have considered 3 classes. If the file is a raw file, then we will do a pre-processing, which consists in labeling the data of the input file that we store in the variable $data$ (line 3). This pre-processing consist to give label x for non sensitive class, y for low sensitive class and z for highly sensitive class. Next, we browse each element of $data$ (line 4): if the label assigned to the D element of $data$ is x (line 5), then the D element is added to the non-sensitive class (line 6); If the value associated with the label is y (line 8), the D element is encrypted with a 192-bits key of the AES algorithm and added to the low sensitive data class (line 9); In the opposite case (the value of the label is neither x nor y but z, line 10), the D element is encrypted with a 256 bits key of the AES algorithm and is added to the highly sensitive class.

Algorithm III.1 Data classification

```
Function DataCategorisation ( File , classNumber)
Begin
    data := Data_labelisation ( File )
    For (each element D in data )
        If ( class_label (D) = x)
            Add D in non_sensitive
            If ( class_label (D) = y)
                Add encrypt (AES192, D) in low_sensitive
            Else
                Add encrypt(AES256, D) in high_sensitive
            endIf
        endIf
    endFor
End
```

In order to optimize the security of patient files, an encryption key can be used for each file separately for a period. Thus, the knowledge of a key for a less sensitive file will not lead to the knowledge of the key for another file. These keys will be generated periodically to have a security strength. This could be an area for research.

3.2 Security during data exchanges

The security of the exchanges is done in four phases (step). The detailed description of the different steps is presented in figure 3.

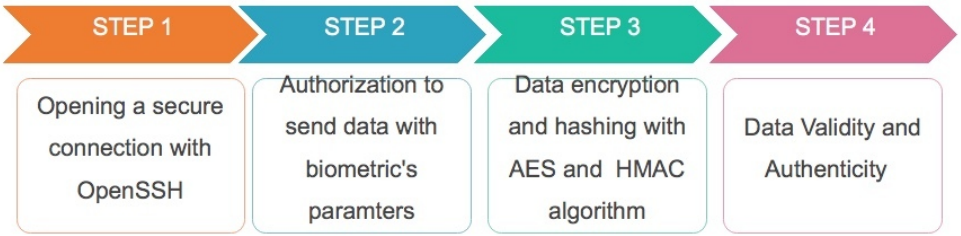

Figure 3: Model's steps

3.2.1 Step 1: Opening a secure connection

This step consists of creating a secure connection to the server to which we want to transfer the data. Several protocols have been created among which Secure Socket Layer (SSL), SSH and others. Unlike SSL which is limited to the HTTP protocol, Secure Shell (SSH) takes into account several communication protocols. We use here SSH connection because of its diversity in taking into account the different protocols.

3.2.2 Step 2: Authorization of data sending

The data theft observed in [11] usually occurs as a result of an intrusion into the system or negligence on the part of the user. For example, if a user logs in to his account, leaves his session open and moves, it is possible that a colleague or a third party uses his session to transfer data to a shadow server. But also, a malicious person can infiltrate the system and steal a lot of data. We have a hacker who infiltrates the system and transfers the data freely to his

personal server. No checks are made and before you know it, the hacker has already committed his crime.

An authentication before sending any data will be necessary to overcome this problem. This authentication will also generate an electronic signature to attach to the data and authenticate the user who transmitted it. This step therefore consists in identifying the user who wishes to transmit the message, but also in authenticating the latter. The authentication is done by using the biometric parameters of the user when he wants to transfer the information. The transfer is conditioned by the biometric parameter, in order to be sure that the user has the right to transfer the patient's file. Authentication is therefore the trigger for a data transfer.

The model described in [**kouam2021**] presents a problem: indeed, the biometric authentication is requested after data extraction. This assumes that the hacker has already had access to the data server and if he cannot transfer it, he can instead copy the data to an external medium or take image captures. Moreover, the model presented in [**kouam2021**] is limited to preventing fraudulent uploads without reporting them. Therefore, it will be important to prevent the malicious person from having access to the data. For this purpose, the medical record is extracted after successful authentication of the user as shown in the figure 4.

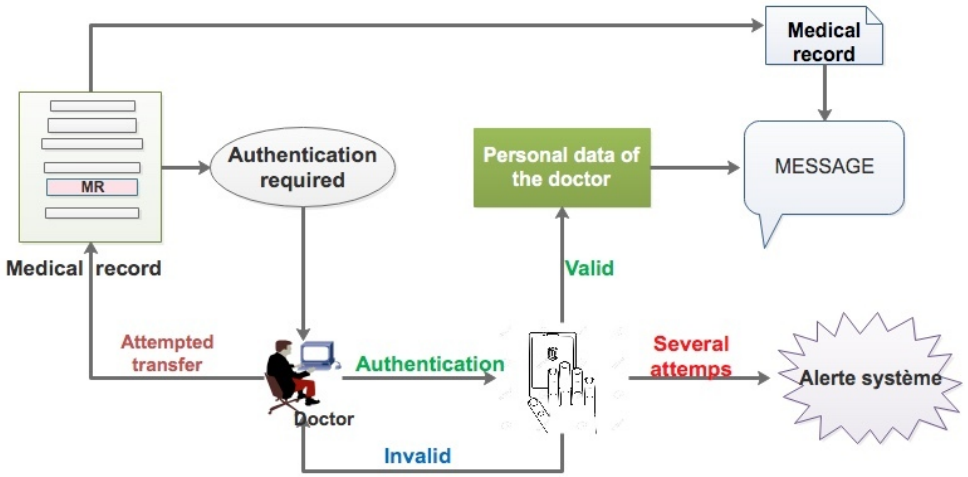

Figure 4: User authentication before data sending

After issuing the request to transmit an EMR, the biometric module is triggered asking the EMR to authenticate itself. If authentication is successful, then the EMR is extracted and paired with the generated signature to form the message that will be transferred. When several authentication attempts fail, the alert system is raised.

Figure 4 can be describe by algorithm III.2 with authentication function which consider in parameter a $File$ and $User$.

First of all, the user must log in to the system in order to gain access. It is therefore assumed that this authentication is not a problem (line 3). If the user wants to make a data transfer (line 4), he need to do a biometric authentication (line 6). In this case, we use in the loop while two variables $compt$ and a (line 5), where a represents the number of connection attempts and $compt$ allows to check if the number of attempts is exhausted. If the user has not yet exhausted the number of attempts and if the biometric parameters retrieved in line 6 are correct (line 7), then the information to be transferred is extracted (line 8), the signature that will authenticate the information is generated (line 9), the message is constructed using the function $Construct(data, param)$

Algorithm III.2 Authentication for message sending

```
1  Function Authentication (File, User)
2  Begin
3      User_authentication_system ( )
4      If (user_action_sending = True)
5          While (compt < a)
6          auth := biometric_authentication (User)
7              If (auth = 1)
8                  data := Data_extraction (id_data, File)
9                  param := User_signature_extraction (User)
10                 MESSAGE := Construct (data, param)
11                 Send (MESSAGE)
12                 Return Message sending
13             Else
14                 compt := compt + 1
15             endif
16         endWhile
17             System_alert ()
18     endIf
19 End
```

that allows to carry out the encryption and hash operations of the message (line 10) and is sent (line 11). If the biometric parameters are not correct (line 13), the counter is incremented (line 14). When the number of attempts is exhausted (exit of the while loop, line 16), a system alert is launched (line 17).

This algorithm is only used to check and detect attackers who try to steal data, but not to track them. In order to intercept the hacker, we introduce deception into the data transfer process by creating a fake local data server. Thus, when an intruder is detected, the alert system is lifted and he is redirected to the fake data server. He will believe he is copying the data and anyone who has access to this server can be tracked. Figure 5 highlights the one stage deception architecture when an intruder gains access. Thus, the model not only prevents fraudulent uploads, but also implements deception. It can be describe by algorithm III.3.

Algorithm III.3 Deception transfer

```
1  Function Deception_transfer (User)
2  Begin
3      If ( biometric_authentication (User) = False)
4          Redirect user in fake medical record
5          System_alert (User)
6          System_detection (User)
7          Behavior_collection (User)
8      Else
9          Message := encryption ( Data_extraction (), param)
10         Sending (Message)
11     endIf
12 End
```

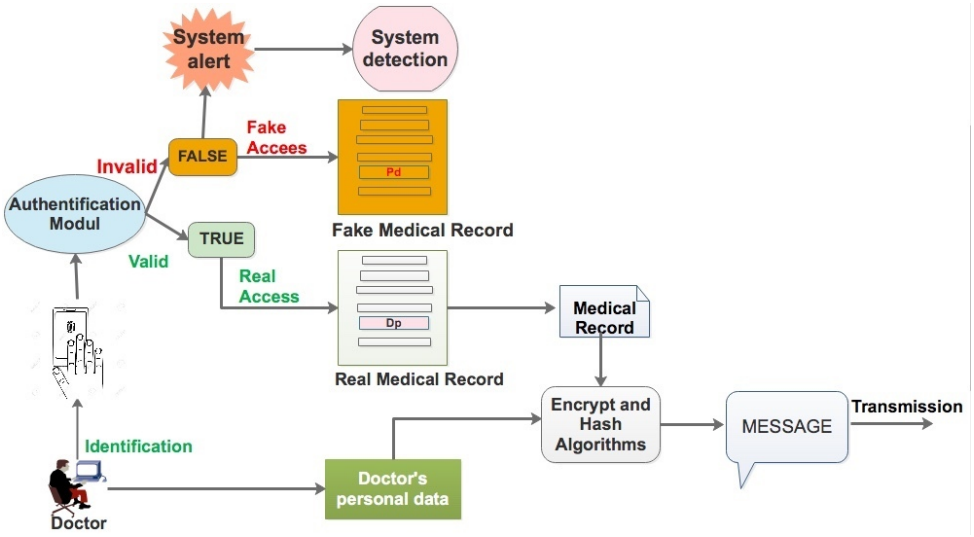

Figure 5: Deception transfer

Following the algorithm III.2, when the user exhausts the number of authentication attempts (line 3), instead of sending him back, he is redirected to an environment with fake or already outdated data (line 4). Then, an alert notification is sent to the administrator (line 5). The attacker is tracked (line 6) and information about his actions is collected (line 7) in order to reinforce system security and anticipate possible attacks. If the authentication is successful, the user is redirect to the real environment and the message is sent (line 9 and 10).

3.2.3 Step 3: Data encryption and hashing

Confidentiality and data integrity are two key services in computer security. Given the sensitivity of the data handled by healthcare systems, professional secrecy and the risk incurred by the modification of a value that could cost a human life, it is important and necessary to be reassured that the transferred data has not been altered and that no one has become aware of its content. Cryptographic and hashing algorithms allow to ensure these two services.

In order to guarantee confidentiality, Rijndael's AES algorithm has been used. This algorithm is used with a 256-bit key to guarantee optimal security because the longer the key size, the more difficult it is to retrieve the message in clear text. Regarding the integrity of the data, the use of the hash algorithm which is a one-way algorithm allowed us to be sure that the message received has not been modified during its transfer. HMAC is a powerful algorithm allowing to generate a signature to verify the integrity. This algorithm has been chosen after a long study that has shown that they are still the best encryption algorithms on the web.

The message sent can be expressed by the equation 1 below:

$$M = E_k(d + s) + H_{kh}(E_k(d + s)) \tag{1}$$

Where:

- **M** is the final message to be sent;
- **d** is the patient's medical record information;
- **s** is a digital signature generated from the doctor's information which will allow to validate the source of the message;

- **E** is the cryptographic function used to encrypt the message;
- **k** is the cryptographic key;
- **H** is the hash function;
- **kh** is the key used by the hash function.

Thus, the total massage to be sent consists of the encrypted main information ($E_k(d+s)$), as well as the hash function associated with this information ($H_{kh}(E_k(d+s))$).

Let us assume:

- $D = d + s$ which is the general information, and
- $m = E_k(D)$ which represents the encrypted message.

The equation 1 becomes the equation 2 below:

$$M = E_k(D) + H_{kh}(E_k(D)) = m + H_{kh}(m) = m + h \tag{2}$$

With $h = H_{kh}(m)$

The message $M = m + h$ is thus sent by the source. When the m message is received, the system will try to decrypt it in order to find the information sent. But first, it will do an integrity text to make sure that the message has not been modified during the transfer process. In order to verify the integrity of the received m message, the system will compute h' in the same way as h and will evaluate the ratio defined by the equation 3:

$$i = \frac{h}{h'} \tag{3}$$

Thus, integrity I is define by:

$$I = \begin{cases} True, & if\ i = 1 \\ False, & otherwise \end{cases}$$

If the integrity value is false, then the message is immediately rejected without the need to decrypt it. Otherwise, the information m is decrypted according to the equation 4 below:

$$m' = D_k(m) = d + s \tag{4}$$

where D is the decryption function and k the decryption key. Once the message is decrypted, we check that the signature is a recognized signature in order to validate the source of the message. So, we define the s function as follows:

$$s = \begin{cases} True, & message\ m\ can\ be\ accepted \\ False, & message\ m\ will\ be\ rejected \end{cases}$$

Indeed, the validation and acceptance of each m message will go through all these tests. This ensures with a certain level of certitude that the accepted and validated messages are indeed authentic and are not the result of an intrusion.

3.2.4 Step 4: Data Validity and Authenticity

Data transferred over the network can come from many sources. Therefore, the authenticity of the received data must be ensured in order to accept it. The authentication solution is proposed above (step 2). It will also be necessary to be able to authenticate a message when it is received to be sure that it comes from the desired source. For example, if an attacker tries to construct a fraudulent file with the initials of a staff member and send it to the server, we must be able to identify and reject the message upon arrival.

The HMAC function can be used in to verify the integrity is a dual role function: it ensures both the integrity and the authenticity of the message to which it is applied. This function makes use of a secret shared between the sender and the receiver of a message. This secret allows the receiver to authenticate the message he has received. We can summarize the process in algorithm III.4.

This algorithm allows to verify the authenticity of the message received by the server or the receiver. When a user wants to make a transfer, the message is constructed (line 3 to 8) and is sent to the destination. At the destination, the received message (line 10) is decomposed (line 11) and the hash is generated (line 12). A test is done to check if the hashed message at the time of sending corresponds to the hashed message at the reception (integrity check). If both values are the same (integrated message, line 13), the signature of the received document is generated to make sure that it corresponds to a known signature (line 14): If the signature is correct (authenticity of message check, line 15), then the message is decrypted and validated (line 16); If not (the signature is not correct, line 17), a system alert is launched (line 18 and 19). If on the other hand the generated hash is not valid (line 21), the message is not even decrypted and a system alert is returned exposing the integrity compromise of the received message (line 22 and 23).

Algorithm III.4 Partitioning function of a sorting algorithm.

```
Function Fraudent_detection (data, User)
Begin
    If (client_computer)
        crypt := encrypt(data, key)
        hash := Hash(crypt, key_h)
        signature := User_signature_extraction (User)
        Message := construct (crypt, hash, signature)
        Sending (Message)
    Else
        message := Receiving (Message)
        Hash, data := Treat (message)
        hash_r := Hash(crypt, key_h)
        If (hash_r = hash)
            signature := Signature_verification (decrypt)
                If ( signature )
                    Return decrypt(data)
                Else
                    System_alert( )
                    return Failure signature authentication
                endIf
        Else
            System_alert ()
            Return Compromise integrity
        endIf
    endIf
End
```

As presented in equation 1, we can model the message sent by an attacker by the equation 5:

$$M_a = E'_{ak}(d' + s') + H'_{akh}(E'_{ak}(d' + s')) \tag{5}$$

Where:

- M_a is the message constructed by the attacker and to be sent to the remote server;
- **d'** is the information that the attacker wishes to transmit;
- **s'** is a digital signature generated by the attacker. It is assumed that the attacker knows the general format for constructing a message;
- **E** is the cryptographic function used to encrypt the message;
- **ak** is the cryptographic key used by the attacker;
- **H** is the hash function;
- **akh** is the key used by the hash function of the attacker.

Assuming the same assumptions as before, we obtain the equation 6:

$$M_a = E'_{ak}(D') + H'_{akh}(E'_{ak}(D') = m_a + h_a \tag{6}$$

When the message is received, as before, many tests will be done successively to validate and accept the message. For the message M_a to be accepted, it will be necessary to make integrity test define by $i' = \frac{h_a}{h'_a}$. For i to be equal to 1, two conditions must be met:

- The hash functions used must be the same: we must have $H' = H$.
- The keys used by the cryptographic function must be the same: we must have $kh = akh$.

If one of the previous conditions is not met, then the message will be rejected immediately. The strong condition is to have $kh = akh$. If these conditions are met, the system will try to decrypt the m_a message by doing:

$$m'_a = D_k(m_a) \tag{7}$$

Two other conditions must be met:

- The decryption algorithm D must match the encryption algorithm E' used by the attacker.
- The key k used for decryption must be able to decrypt an encrypted message using the key ak used by the attacker.

Indeed, if one of these conditions is not met, this will have a direct impact on h'_a because the latter is generated using m_a. In addition to these conditions, we will check that the signature s' is a signature recognized by the system to finally validate the message received. If the signature is not recognized we will know that the message comes from an attacker.

Thus, the equation of construction of h is very interesting because it depends on the following parameters

- The encryption algorithm;
- The encryption key;
- The hash algorithm;
- The hash key.

Indeed, if one of these parameters is wrong, the message will be rejected.

The electronic signature generated when the message is authorized will identify the sender of the received message. If this signature is not recognized, even if the authenticity of the message is verified, the latter will be rejected and an alert will be launched for a high-level intrusion. Thus, not only the received data is authenticated, but also the sender of the message.

3.3 Model summarization

Our general idea is to propose a multi-stage deception for data security in information systems as presented in figure 6. This model is composed of three modules: module 1 which allows to manage the access to the system, module 2 which emphasizes the first level of deception and the third module which emphasizes the second level of deception.

The model distinguishes several states in which a user may find himself, including the black state for hackers, the gray state 1 for any user who has had proper authentication, the gray state 2 for users who have gone through module 1 (first level of deception) and finally the white state for users who are compliant with the system rules. Each module implements control and verification algorithms to move a user from one state to another.

The work presented in this article covers module 1 and a part of module 2. Module 2 and 3 will be present in our next article

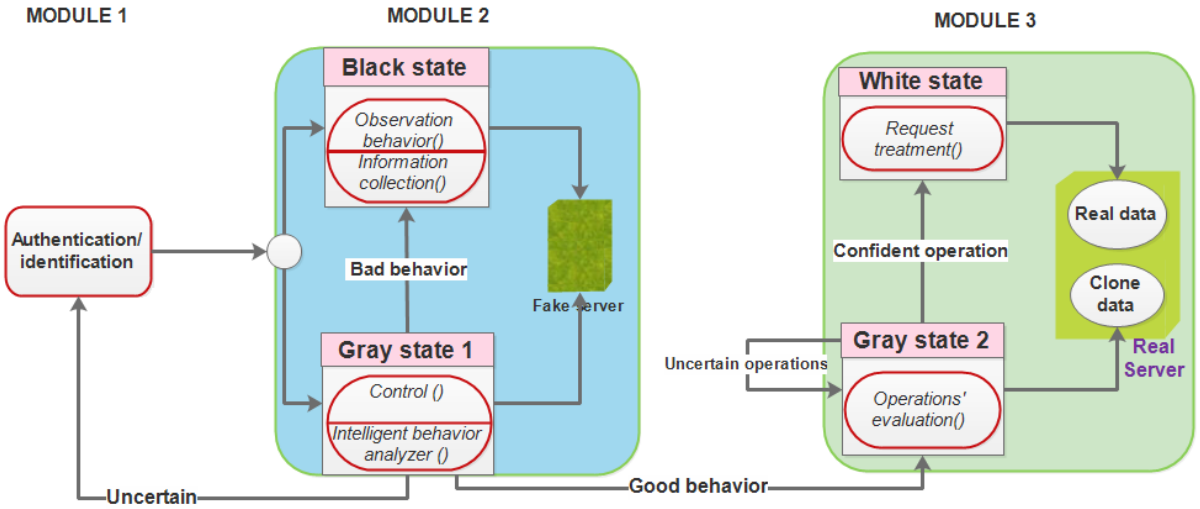

Figure 6: General model

IV RESULTS AND DISCUSSIONS

4.0.1 Local security

This work was simulated in the Bahmni health system deployed in several health centers in Cameroon. This system is composed of three modules among which the OpenMRS module which manages the clinical and patient management part.

The first results were obtained on local security. The dataset using during test is database of OpenMRS available in [16]. This database contains a maximum size of 5000 patients with almost 500,000 observations. In order to better observe the choice made for local security, the database was loaded with different sizes. We considered a database containing 100, 500, 1000, 2000, 3000, 4000 and 5000 patients.

To highly our result, we also implement Akinyele at al [3], Basel Katt [8] and Mohammed MIROUD's work [10] method. [3] and [8] consider a data base as a file and encrypted the whole: we will call it global encryption. [10] classify data and three classes and encrypt each whole class which different size of AES algorithm: we will call it total encryption. Our propose model is called partial encryption.

When these methods are applied to a single piece of information, they are all almost equivalent because the difference in observed time does not seem to be considerable. But since a server is made to store large amounts of data and allows multiple accesses, we will consider a simulation assuming that 100 users try a simultaneous access to the system. This simulation will consist in running in a loop of 100 iterations an access to an information of the server by each method. We will therefore recover the time taken by each approach according to the volume of data on which the algorithm is applied. The function executed by each method is presented by the algorithm IV.1.

Running this algorithm (algorithm IV.1), we obtain the results presented in the table 1 presenting each method of the literature described previously as a function of data volume and execution time.

From table 1, we can represent a curve (Figure 7). This curve allows to see the real evolution of time of the three experimented methods according to the amount of data.

Algorithm IV.1 Test

```
Function Global encryption(File F, ID)
Begin
    For i from 1 to 100 do
        F1 := Decrypt(F)
        X := Search (F1, ID)
        Encrypt (F1)
    endFor
    return X
End
Function Total encryption(File F, ID)
Begin
    For i from 1 to 100 do
        Class1 := Decrypt128 ($F_1$)
        Class2 := Decrypt192 $(F_2$)
        Class3 := Decrypt256 ($F_3$)
        X := Search (Class1, ID)
        Y := Search (Class2, ID)
        Z := Search (Class3, ID)
    endFor
    return X, Y, Z
End
Function Partial encryption (File F, ID)
Begin
    For i from 1 to 100 do
        X := Search ($F−1$, ID)
        $Y_1$ := Search ($F_2$, ID)
        $Z_1$ := Search ($F_3$, ID)
        $Y := Decrypt192(Y_1);$
        $Z := Decrypt256(Z_1);$
    endFor
    Return X, Y, Z
End
```

Table 1: Execution time depending on the volume of data and the method. The table show time encryption on different volume of data using different methods: Global encryption which treats the database as a file and encrypts the whole; Full encryption which treats the database as a file and encrypts only the useful information contained in it; Partial encryption which partitions data before encryption

Patient's number	Global encryption	Total encryption	Partial encryption
100	116.497	7.065	0.395
500	148.879	34.171	0.415
1000	181.317	68.757	0.424
2000	250.765	135.325	0.427
3000	314.756	199.272	0.476
4000	391.401	268.228	0.481
5000	468.772	332.001	0.553

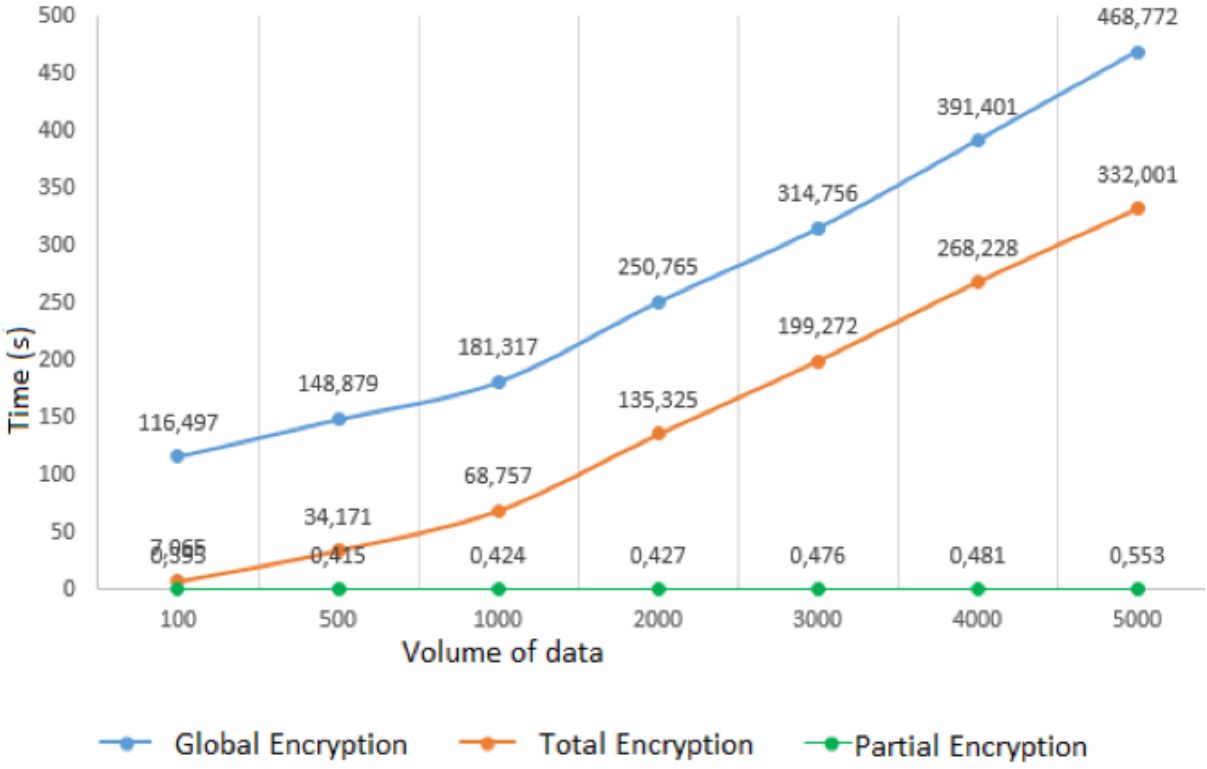

Figure 7: Representative curve of data access time as a function of data volume

According to the curves presented by the figure, we realize that with global encryption, which is the method proposed by Akinyele at al [3] and Basel Katt [8], the access time is proportional to the volume of data. This can be explained with the fact that the time to decrypt data depends on its volume and the size of the key. Since the model uses the same key to ensure encryption and all the data is encrypted in one block. The access to an information will require decryption and encryption of the whole information. The model proposed by Mohammed MIROUD [10] is better than those proposed by [3] and [8] with the fact that the latter considers several classes of data. If the volume of data is around 100, then the access time seems to be zero. However, the model loses its access time capabilities with increasing data volume because although the data is segmented into several classes, each class is encrypted by block.

As for the partial encryption, the curve does not evolve with the increase of the data volume. This method consists in encrypting each element of each class independently. This allows us to have a real time access to the system whether it is in access or in storage.

4.0.2 Security transfer

To test our data transfer model, we made the data transfers using the OpenSSH server to create the secure connection. The information systems face nowadays multiple attacks among others Man in the Middle, brute force attacks, source or destination repudiation, intrusions in the information system and many others. The proposed model will help limit many of them.

On the other hand, the attacker could try to vote the system data or modify it for a specific purpose. To do this, he will try to access the database directly to commit his crime. The encryption performed during the storage of the informations (figure 2) will make them unidentifiable for the hacker. Indeed, if it is a specific data theft, the latter will not be able to identify the targeted

information. If it is a mass theft, he will have to download a volume of encrypted data that will take time to decrypt. If a particular encryption key is used for each piece of data, then the hacker's knowledge of one key will not allow him to have knowledge of all the keys of the other data. Moreover, when trying to find the key of the data, it will be necessary to consider the classes of data if the hacker wants to have the complete information of each patient file for example.

The model also solves the problem of repudiation of the source because once the signature linked to the sending of the data is validated, we will know where the message came from and who initiated it using signature. One of the problems usually encountered in structures is when one person uses the other's computer to commit malicious actions in order to hide his identity. These actions can also consist in an illicit consultation or a fraudulent sending. The introduced biometrics will allow to fight these types of internal threats generally met. Brute force attack cannot be efficace here if the attacker has access to the server do to the fact that data is classify and attacker need to apply it on each element and each classe to have the clear information.

The partial encryption used for the local security optimizes the processing time in a remarkable way. The access is done almost in real time with the increase of the data volume. This model also appears to be faster than Mohammed's [10] because the keys used are less long for each class of data. We focused not only on security, but also on data availability. Indeed, encrypting the data before storing it has made it possible to make a medical record in the database unidentifiable, but also to make this database unavailable. In addition, encrypting each item individually significantly reduces the time it takes to access the cloud, and in some cases can also allow for direct search in the cloud.

With data theft in computer systems increasing day by day, an effective measure is needed to deal with this. The use of the encryption algorithm ensures the confidentiality of the exchanged data. We have used the AES algorithm which is one of the safest algorithms nowadays. As for the integrity and authenticity of the received message, we used the HMAC-SHA2 function which is also one of the most secure functions.

The key process of the model presented above is the authentication of the sender before sending the message and the authenticity of the received message. The use of biometric parameters is an efficient and very fast way to authenticate a user so that he does not have to enter a password every time he sends a message. This biometric parameter serves two purposes: first of all to control the data exchanges, i.e. to protect the data transfer module, but also to authenticate the source by identifying the sender in order to avoid the repudiation of the source. Indeed, the HMAC function used here, which uses a shared secret, in addition to guaranteeing the integrity also guarantees the authenticity of the received message. Thus, the coupling of HMAC and the biometric parameter is a strong element for the authenticity of the received data. Table 2 show a comparison between our propose and existing literature method.

From table 2, 1 (one) mean the technique is use and 0 (zero) when method is not use. We can see that our propose is the best more than literature model and can improve high quality of security.

Contrary to the literature, many attacks can be addressed by the proposed model such as attacks directly targeting the data, i.e., transmitting fake messages. We can list among others Man-in-the-Middle attacks, password attacks, identity theft, birthday attack, phishing, etc. Indeed, for a message to be accepted, many parameters must be considered such as the encryption key, the digital signature of the sender and the key used by the hash function.

Table 2: Comparaison between our propose technique and existing technique. We make it through different methods in security

Authors	Encryption	Hash	Signature	Biometric
Akinyele at al [3]	1	0	0	0
Basel Katt [8]	1	0	0	0
Rao [4]	1	0	0	0
Katsikas and Al [2]	1	0	1	0
Mohammed MIROUD [10]	1	1	1	0
Our model	**1**	**1**	**1**	**1**

The birthday attack, although effective in bypassing the integrity of a message and substituting the original message with its own message, will be ineffective against the proposed model, because the attacker will also have to find the encryption key and the electronic signature that is required for the message to be accepted.

Our model also incorporates the multi-stage deception principle by redirecting the attacker to a fake server or fake data to detect and apprehend him.

4.1 Conclusion

This article addresses the problem of data security in healthcare systems. The objective was to propose an effective security model for data protection in healthcare systems while introducing the concept of cyber security. This article addresses the confidentiality, integrity, authenticity, non repudiation of data as well as the availability of data. In addition to that, the work helps to detect intruders in the proposed system.

The proposed model is composed of two modules: a first module to ensure local security and a second module to ensure data transfer. Concerning the local security, we proposed a solution of partial encryption of the data consisting in partitioning the data in three classes namely non-sensitive, low-sensitive and hight sensitive which were stored in clear, encrypted with a cryptographic key of 192 bits and a key of 256 bits respectively. This allowed us not only to strengthen the level of local data security, but also to ensure real-time availability regardless of the volume of data. Regarding transfer security, the use of encryption and hash algorithms allows us to ensure cryptographic services such as confidentiality and integrity. The addition of the doctor's signature at the time of sending ensures the authenticity of the message at its reception. By calculating the hash of the encrypted message containing the digital signature, we realize that the hash function considers several strong parameters.

In addition to that, the use of biometrics when sending data allows to solve the problem of fraudulent sending within the institution. Indeed, these biometric parameters are also used to generate the digital signature of the doctor, which makes it totally unique and difficult to fake. The introduction of cyber deception allows us to redirect users to a deception environment in order to study their behavior and actions in the system, which will then allow us to strengthen the security of the system. This allows us to save time, but also to push the attacker to stop looking for security holes.

The results obtained allowed us to validate the different modules proposed, but also to note that the introduction of biometrics for the control of critical actions in the information systems will make the security more robust. The proposed security model ensures good data security, but does not prevent the system against the various attacks encountered in the field of cyber

security such as denial of service, spam, spoofing, etc.. The advantage of this system is that it is no longer possible for a user to deny an action he has taken. The disadvantage of such a system is that it becomes difficult to hand over a task to a colleague with the consideration of biometrics. Moreover, if a user becomes unavailable, it will take time to configure a new user who can take over the tasks of the latter.

In this article, we have just introduced a deception environment. This environment to be efficient must use a fake data set to feed the server in the environment, which will be manipulated by the redirected users. Therefore, our next work will be focused on the generation of data in the context of cyber deception in order to feed the servers in this environment.

REFERENCES

Publications

[1] N. LENOIR. "la loi 78-17 du 6 janvier 1978 et de la commission national de l'informatique et des libertés : ELEMENT POUR UN PREMIER BILAN DE CINQ ANNEES D'ACTIVITE". In: *La revue administrative* 36.215 (1983).

[2] S. K. Katsikas, D. D. Spinellis, J. Iliadis, and B. Blobel. "Using trusted third parties for secure telemedical applications over the WWW: The EUROMED-ETS approach". In: *International Journal of Medical Informatics* 49.1 (1998), pages 59–68.

[3] J. A. Akinyele, M. D. Pagano, Z. N. Peterson, C. U. Lehmann, and A. D. Rbin. "Securing electronic medical records using attribute-based encryptionencryption on mobile devices Report 2010/565". In: *proceeding of the 1st ACM workshop on Security and privacy in smartphones and mobile devices*. ACM. 2011.

[4] M. A. Leena and R. Kakoli. "Centralized Database Security in Cloud". In: *International Journal of Advanced Research in Computer and Communication Engineering, I* 8 (2012), pages 50–68.

[5] E. M. Mohamed, H. S. Abdelkader, and S. El-Etriby. "Enhanced data security model for cloud computing". In: *2012 8th International Conference on Informatics and Systems (INFOS)*. IEEE. 2012, pages CC–12.

[6] S. Biedermann and S. Katzenbeisser. "POSTER: event-based isolation of critical data in the cloud". In: *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. 2013, pages 1383–1386.

[7] C. Quantin, F.-A. Allaert, B. Auverlot, and V. Rialle. "Sécurité, aspects juridiques et éthiques des données de santé informatisées". In: *Informatique médicale, e-Santé*. Springer, 2013, pages 265–305.

[8] B. Katt. "A Comprehensive Overview of Security Monitoring Solutions for e-health Systems". In: *2014 IEEE International Conference on Healthcare Informatics*. IEEE. 2014, pages 364–364.

[9] Y. Sun, J. Zhang, Y. Xiong, and G. Zhu. "Data security and privacy in cloud computing". In: *International Journal of Distributed Sensor Networks* 10.7 (2014), page 190903.

[10] M. E. M. Miroud. "la sécurité dans les systemes de e-santé". PhD thesis. Université des sciences de la technologie d'Oran Mohamed Boudiaf, juin 2016.

[11] B. Edward. *Les données de 500.000 patients en france publier sur le dark web*. https://www.futura-sciences.com/tech/actualites/cybersecurite-donnees-medicales-500000-francais-vendues-dark-web-85924/, Accessed: Febuary 25, 2021. Feb. 2021.

[12] I. G. Kouam Kamdem and M. J. A. Nkenlifack. "Data Security in Health Systems: Case of Cameroon". In: *Intelligent Computing: Proceedings of the 2021 Computing Conference, Volume 3*. Volume 285. Springer for Lecture Notes in Networks and Systems. 2021, pages 48–57.

[13] L. Parisien. *Victime d'une attaque informatique, l'hôpital de Villefrache-sur-Saône contraint de déprogrammer des opérations*. https://www.ouest-france.fr/societe/cyberattaque/rhone-l-hopital-de-villefranche-sue-saone-victime-d-une-attaque-informatique-7155641, Accessed: Febuary 25, 2021. Feb. 2021.

[14] V. Alice. *Victime d'une cyberattaque, l'hôpital de Dax fonctionne au ralenti*. usinedigital.fr, https://www.usine-digitale.fr/article/des-hackers-multiplient-les-campagnes-de-phishing-bancaire-en-france.N1064899, Accessed: Febuary 25, 2021.

[15] S. Froment. *Mon qualiticien modernise la gestion de la qualité dans les établissements de santé*. https://www.caducee.net/actualite-medicale/14719/mon-qualiticien-modernise-la-gestion-de-la-qualite-dans-les-etablissements-de-sante.html, Accessed: April 10, 2019.

[16] B. Mamlin. *Demo data*. https://wiki.openmrs.org/display/RES/Demo+Data, Accessed 5 July 2020.

V ACKNOWLEDGEMENTS

The authors wish to thank:

- Alexander Von Humboldt Foundation, for his support, in particular the acquisition of part of the equipment and research tools, with the funding received from this foundation in 2019, for the project entitled "Multi-scale Analysis and Data Processing".
- AUF (Agence Universitaire de la Francophonie) for his support, in particular the strengthening of the technical platforms of our laboratories in 2019, with the funding received for the project entitled "Santé Numérique Sécurisée : Analyse et Sécurisation des données Big data pour les prédictions d'intérêt médical".

VI BIOGRAPHY

Igor Kouam PhD student in University of Dschang since 2019. He had a Master degree in Computer science, option Network distributed services. From URIFIA Laboratory, the author is an active member of Health care project in Dschang University.

Marcellin Nkenlifack has been Head of the Department of Mathematics and Computer Science at the Faculty of Sciences of the University of Dschang since 2018, after having been Head of the Department of Computer Engineering at the IUT-FV in Bandjoun from 1996 to 2018. He is a specialist in Services and Connected Objects, holds a Ph.D. in Software Engineering and Automation of Hybrid Systems, and is a Design Engineer in Computer Engineering from the Polytechnic School of the University of Yaoundé 1. He is also a "graduate" of the UNU (United Nations University) in formal methods in Software Systems Engineering. He is an expert on the "ICT and Artificial Intelligence" Commission of the National Technology Development Committee (CNDT). He has been a guest researcher at Institut Galilée -Paris13, SUPELEC -Rennes, Université Cheikh Anta Diop -Dakar, Université Felix Houphouet Boigny -Abidjan, CFICIRAD -Brazzaville. He has successfully piloted numerous research and development projects supported by international organizations such as the Silicon Valley Community Foundation, Agence Universitaire de la Francophonie, and the US-Ar Research Laboratory. He has been a member of the CARI Standing Committee (African Conference on Research in Computer Science and Applied Mathematics) since 2016 as one of the representatives of African researchers.

He has twice received the "Best Instructor of the Regional Academies of Internet and Cisco High Technologies" award for the West and Central Africa and Great Lakes Region (24 countries in total) at the Cisco Networking Academy African Safari International Conference on May 4-5, 2010 in Abuja, Nigeria, then at the July 12-14, 2016 edition in Kinshasa, Congo DRC.