A Lightweight and Secure Framework for Hybrid Cloud Based EHR Systems

Basit Qureshi[1(✉)], Anis Koubaa[1,2,3], and Mohammad Al Mhaini[1]

[1] Prince Sultan University, Riyadh, Saudi Arabia
{qureshi,akoubaa}@psu.edu.sa, malmhaini@gmail.com
[2] Gaitech Robotics, Hong Kong, China
[3] CISTER, INESC-TEC, ISEP, Polytechnic Institute of Porto, Porto, Portugal

Abstract. Recently Cloud based Electronic Health Records Systems have been developed and are being used in the healthcare industry. Albeit the various benefits of the technology, security, trust and privacy are a major concern. In this paper, we present a secure and affordable framework for EHR storage, leveraging the properties of hybrid cloud in securing data in addition to building low power back-end cluster constructed using low cost single board computers (SBC). We detail requirements for a secure cloud based EHR framework and present the system architecture based on the publisher/subscriber model. The framework is developed and tested on a Hadoop based cloud cluster using SBCs as nodes. Efficiency of the framework is measured in terms of response time for various sizes of Data Blocks containing EHRs.

Keywords: Hybrid cloud computing · EHR systems · Encryption

1 Introduction

Over the last few years, cloud computing has grown to be a new service model and has had a tremendous impact on the establishment of several cost-effective platforms for hosting largescale service applications in data centers. Nonetheless, notwithstanding the considerable benefits and services, the issue of security and privacy has been at the forefront of consumer confidence in the availability of confidential information in public domain [1]. There are many Electronic Health Records Management (EHR) systems available, most of these follow a client server model where the service provider maintains data centers hosting data while clients can access these data. The variety and size of these data systems challenge researchers to accurately and effortlessly integrate data from various sources while maintaining integrity, security and availability using reliable channels [4, 8].

Several researchers have concentrated their efforts in developing various architecture for healthcare based applications in the cloud [1–6]. Fang et al. [9] present a survey on Health informatics in the cloud. Authors in [10, 11] investigate privacy preserving secure communication in cloud assisted e-healthcare systems. Researchers in [3, 12] present a security aware cloud enabled eco-system for healthcare data storage and analytics. Shrestha et al. in [4] present a eHealth Framework for security and privacy. Suresh in [5] detail a secured cloud based health record model for storage of

R. Mehmood et al. (Eds.): SCITA 2017, LNICST 224, pp. 197–206, 2018.
https://doi.org/10.1007/978-3-319-94180-6_20

medical records in the public cloud domain. In most of the aforementioned works, the mechanism of security heavily involves cryptographic algorithms and application of various encryption techniques. As pointed out in [1, 2], a major concern in these mechanisms is the quality of service issue; as the size of the records grow, the time to encrypt and decrypt large records increases significantly consequently affecting the latency in data transfer thus affecting the QoS parameters. Another major issue is the search keywords and access control mechanisms where certain users in the EHR system have privileges to access certain parts of data records. In such systems, based on the search keywords, the entire data blocks matching the query results are downloaded, decrypted based on access control parameters and served to the consumer which is very inefficient in terms of performance.

In this work, we address the aforementioned issues by detailing a lightweight framework utilizing hybrid cloud for preservation of security and privacy. The proposed framework takes inspiration from the work presented in [6] vertical partitioning of EHRs to decrease the latency costs incurred in keywords search. The data is encrypted using an access control policy based on credentials. The access control policy is defined as joints and disjoints with various attributes allowing the users with appropriate credentials to have access to the data. The resulting framework is evaluated in the hybrid cloud environment using a Hadoop based cluster. Data is stored in Hbase and served based on queries and access control information. Initial results demonstrate successful implementation of the framework for lightweight data control.

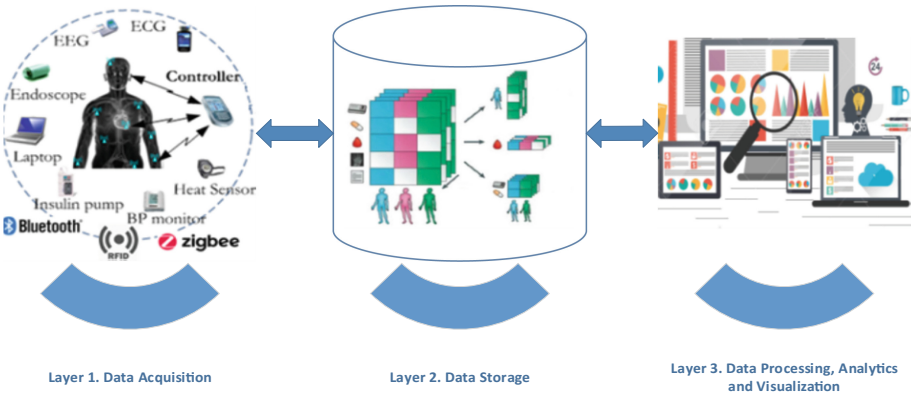

Fig. 1. Layout of 3 layers in the proposed framework

2 Framework Requirements and Design

In this section, we reflect on the requirements for a hybrid cloud based design and architecture of secure and power efficient framework for cloud based Electronic Health Records System. A mainstream cloud based EHR system consists of three layers, (i) Data acquisition, (ii) Data storage in Cloud, (iii) Data Processing, analytics and visualization.

Data acquisition layer collects data in any of the various mechanisms for data entry. It could be as complex as a Wireless Body Area Network (WBAN) [13] consisting of various wireless wearable sensors for specific medical applications such as blood pressure, thermometer etc. These sensors and devices can connect to a data collection device using Bluetooth, Zigbee protocols or similar wireless communication medium, and transmit the information to an intermediate cloudlet for synchronizing and processing information.

Cloud Storage Layer provides a reliable, scalable and secure storage that is perhaps the most important functionality of Cloud based PHR systems. Due to large volumes of data being generated, it is critical to provide a secure, efficient and reliable storage and retrieval system for personal records. To enable searching over the data, the customer must either store an index locally, or download all the (encrypted) data, decrypt it and search locally. Both approaches negate the benefit of cloud computing and are therefore poor choice for processing of large scale data thus increasing the overhead of security and volume of communications.

Processing Data is the third important aspect of cloud based PHR systems. Data analytics requires processing of large volumes of data to be used in decision support for Health care providers. Furthermore, visualization of data can be of much benefit to the stakeholders.

The Health Insurance Portability and Accountability Act (HIPAA) [7] states that data privacy must be protected within every layer of a EHR system. Here we detail the requirements for each layer of the system.

(i) Data acquisition Layer. The acquisition layer in Fig. 1 is composed of Wireless Body Area Network sensor devices. These devices have limited computational capacities and operate on finite battery power. Encryption schemes used to protect the communication within BAN sensors and BAN-to cloudlet communications should not be computationally intensive.

(ii) Data Storage Layer. There can be various stakeholders involved in storage, retrieval and sharing of data from the system including, patient, doctor/nurse, emergency unit staff etc. Conventional encryption techniques may not work and would not be able to handle multiple keys from multiple parties.

(iii) Data Processing, Analytics and Visualization Layer. To process data, it must first be decrypted. Decryption of this data in the intermediate stage requires trusted storage infrastructure possibly at the site of the HCP necessitating a private cloud infrastructure.

Further to the above, in an enterprise environment many stakeholders may have access to various kinds of data. Doctors, patients, nurses, emergency unit staff, medical insurance personnel etc. to name a few, require strict access control mechanisms in place. For any efficient solution, it is imperative to address the above-mentioned issues thoroughly and effectively.

3 Hybrid Cloud Based Architecture

Storage services based on public clouds provide consumers with scalable, reliable and dynamic storage at the cost of security trust and privacy. Privacy critical data such as personal health records need to be secured on the public cloud, whereas data with low privacy requirements such as data dictionaries etc. or intermediary data such as data generated from health analytics or visualization, may not require stringent security measures. Figure 2 shows an overview of the public and private cloud domains of the framework.

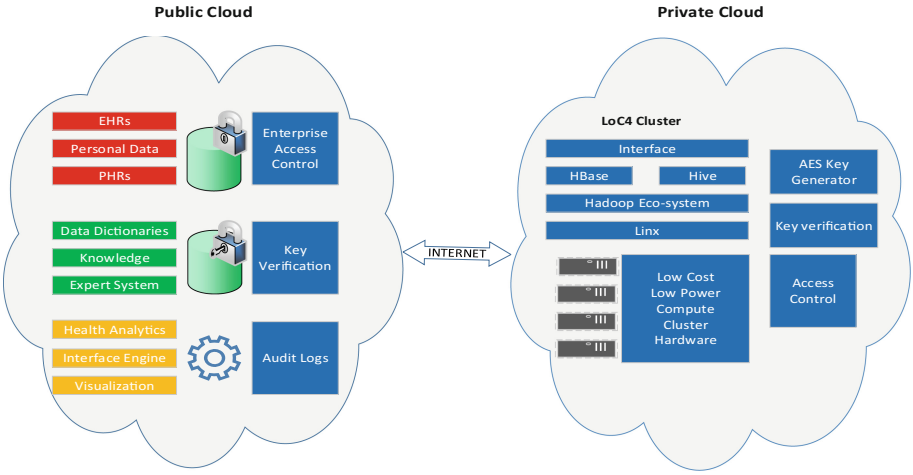

Fig. 2. Hybrid cloud architecture

Large volumes of data generated at the Data acquisition layer needs to be stored and shared with appropriate consumers with valid credentials. Access Control parameters are defined in the private cloud service and are periodically updated in the public service. Public cloud service affords reliable and fast processing of data due to large amounts of compute power available at the service provider. However due to privacy concerns it is not feasible to decrypt data in the public cloud domain, process it and then re-encrypt for further storage. Consequently, a publisher/subscriber model of the system may need to download the encrypted data on a local machine, decrypt, process and upload it to the cloud service after re-encrypting this data. Salient features and components of the proposed publisher/subscriber model is given in Fig. 3. We develop a custom encryption approach that is efficient and reliable and is described further in this section.

The private cloud service essentially hosts a cluster of machines realizing a small data center at the premises of the enterprise. The infrastructure is secured in the enterprise locale with physical, data and network security measures in place. The data center hosts low cost, efficient and eco-friendly Hadoop cluster. Access control information for the enterprise is stored in HBase and Hive data warehouse which are

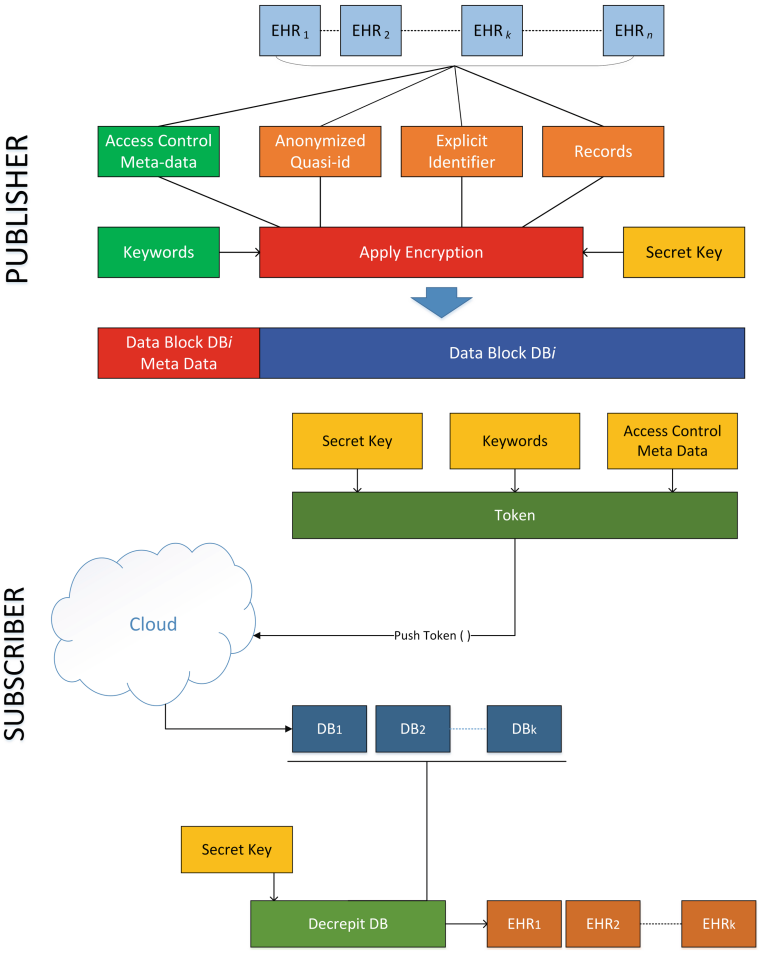

Fig. 3. Publisher subscriber model for encryption of data blocks in the framework

encrypted using commonly available encryption techniques. Additional modules for encryption key generation and verification are implemented in the private cloud. These are necessary for efficient and reliable storage and sharing of data. Communication takes place through the interface which implements a light weight communication protocol for data transmission between the public and private cloud services.

Communicating devices must agree on a secret-key before using AES encryption by using generic key exchange algorithm such as Diffie-Hellman (DH) [14]. Irrespective of the kind of encryption scheme, a consumer must agree on key(s) to encrypt/decrypt messages with a service provider. Advanced Encryption Standard (AES) is one of the most widely used symmetric key encryption algorithms and is accepted as an industry and a government applications standard. AES is optimized for speed, low memory footprint and energy efficiency. Data is encrypted using symmetric

encryption scheme (e.g., AES) under a unique key such that a search index is encrypted so that its contents are hidden except to a party that is given appropriate tokens. The token can be generated with the knowledge of a secret key such that the retrieval procedure reveals nothing about the files or the keywords except that the files contain a keyword in common. This light weight technique addresses the inefficiency of downloaded an entire encrypted block and processing only a limited part of it. Furthermore, the coupling of access control mechanisms allows reliable sharing of appropriate data with proper stakeholders.

To implement this security mechanism, we define a light weight technique inspired by the work presented in [6]. The data is encrypted using an access control policy based on credentials. Only the user whose credentials satisfy the access policy can have access to the data. The attributes can be the profession (e.g., Doctor, Nurse) or the department (e.g., Cardiology, Emergency) of the user. An access policy can be defined as joints and disjoints with various attributes such as

`(Doctor AND Pediatrics) OR (Nurse OR Intensive-Care-Unit)`

which gives access to a Doctor in Pediatrics Department or a Nurse or an Intensive-Care-Unit staff member.

We define two algorithms *PushDataBlock* and *PullDataBlock*, that control read/write of data blocks using a combination of secret key and access control meta data, in the public cloud service as can be seen in Fig. 4. In the following text, we describe the working of these two algorithms to enable secure data communication and storage of a data block in the public cloud service.

algorithm ***PushDataBlock***
input: DataBlock B_i, SecretKey S_k, AccessControlMetaData AC_{MDi}
output: Encrypted DataBlock EB_i
for $i = 1$ **to** $n - 1$ **do**
KeyGenerator(B_i, S_k, AC_{MDi})
PushCloud(EB_i)
end

algorithm ***PullDataBlock***
input: SecretKey S_k, *Keyword*
output: Decrypted DataBlock DB_i
for $i = 1$ **to** $n - 1$ **do**
RequestToken(AC_{MD} , S_k, *Keyword*)
AcquireToken(S_k)
PushToken(*token*, S_k)
PullDataBock(DB_i)
VerifyDataBlock(DB_i)
end

Fig. 4. The PushDataBlock and PullDataBlock algorithms.

Each subscriber is allocated a Secret Key S_k to facilitate access to the private cloud service, these keys are managed by the administrators at the enterprise and assigned to stakeholders with appropriate access control credentials.

Assuming a data block B_i was generated (data acquired from WBAN cloudlet etc.) consisting of one or many records/files and is ready to be uploaded to the public cloud storage, the publisher would call the PushDataBlock with an access control meta data AC_{MDi} for this data block B_i.

The *KeyGenerator* is responsible for properly indexing and encrypting the data block. KG is responsible for encrypting various data blocks using access control information provided and appends metadata for identification of datablocks such as timestamp, size, keywords, ownership etc. It encrypts the data block in such a way that given a *token*, a subscriber can retrieve pointers to the encrypted records/files that contain a *keyword*. Without appropriate *token*, the data block would not be decrypted. The tokens cannot be generated without a secret key S_k. It must be noted that the retrieval procedure does not reveal any details within the data block except only the searchable keywords. Once the *KeyGenerator* has encrypted the DataBlock B_i, *PushCloud*(B_i) is called to push the data block to the public cloud storage.

When a subscriber intends to access a data block for processing or sharing, it needs to call the *PullDataBlock* with the secret key S_k, and the access control meta data AC_{MD}. *RequestToken* call is made for a search keyword term to the private cloud service. Based on the access control policy, the *AcquireToken*(S_k, AC_{MD}) generates a token which is then sent to the public cloud service using the *PushToken*(*token*). The public cloud service uses the token to find the appropriate encrypted documents. The *PullDataBlock* call returns the selected documents as data blocks to the subscriber. At any point and time, the subscriber can verify the integrity of the data blocks by calling the *VerifyDataBlock* call.

A prototype of the proposed framework is built and successfully tested. The next section presents experimental results.

4 Results

The implementation of the proposed framework is based in a SBC based cluster reported in [15]. This cluster is composed of 40 nodes consisting of Raspberry Pi Model 2B and Odroid Xu-4 computers. All nodes are interconnected using gigabit Ethernet. Hadoop 2.6.2 is installed on all nodes with additional installations for HBase and Hive. To optimize the performance of the RPi Cluster, yarn-site.xml and Mapred-site.xml were configured with 852 MB of resource size allocation. The Master node is installed on a regular PC running Ubuntu 14.4 and Hadoop. The default container size on the Hadoop Distributed File System (HDFS) is 128 MB.

A light weight middleware is written using bash script which is responsible for executing *KeyGenerator*, *PullDataBlock*, *RequestToken* and *AcquireToken*. The middleware is executed taking various parameters including EHR (as xml File of various

sizes) along with associated Meta-Data for the file. Access Control parameters were generated for four users (Doctor, Nurse, Manager and Patient). We use AES for encryption of records before it is stored in the cluster. All operations including encryption and decryption time, cost of key generation are linear to the number of attributes. For this experimentation about 100 DataBlocks were pre-loaded in the cluster composed of various EHRs (as xml files). The middleware is executed to simulate the subscriber aspect of the framework with various request sizes and access control parameters. We measure the response time of the *PullDataBlock* request in terms of milliseconds as shown in Table 1.

Table 1. Response time for PullDataBlock request

Number of parallel requests	Total DataBlock(s) size in MB	Response time in milliseconds
1	1.8	6009.9
2	3.6	6101.2
3	5.1	6108.1
5	9.7	6204.8
8	14.9	6208.0
10	18.2	6410.0
15	37.8	7022.6
20	51.7	9514.0
30	108.4	11089.1
50	394.6	28223.1
70	728.4	42188.2
100	1352.5	92579.5

Our results show that the Framework is successfully implemented with reasonable response times from the Hadoop cluster. It must be noted due to the settings in the SBC based cluster the minimum response time for executing Hadoop jobs is approximately 6 s, given these conditions the response time for smaller workloads i.e. less than 50 MB is within one second. As the size of the DataBlocks increases, the time also increases linearly.

5 Conclusions

In this paper, we presented a secure framework for EHR storage based on a hybrid cloud computing architecture. We detail requirements for a secure cloud based EHR framework and present the system architecture based on the publisher/subscriber model. The framework is developed and tested using Hadoop based cloud cluster using SBCs as nodes. Efficiency of the framework is measured in terms of response time for

various sizes of Data Blocks containing EHRs. Results show successful deployment of the proposed framework on the testbed. We intend to extend the implementation of the framework in the public cloud domain which has not been completed as of now. We intend to develop Microsoft Azure web services for the publishers' part of the framework. Both parts of the framework would be thoroughly tested.

Acknowledgements. This work is partially funded by the Robotics and Internet of Things Unit (RIoTU) at Prince Sultan University.

References

1. Yang, J.J., Li, J., Niu, Y.: A hybrid solution for privacy preserving medical data sharing in cloud computing. Future Gener. Comput. Syst. **43**(44), 74–86 (2015)
2. Manoj, R., Alsadoon, A., Prasad, P.W.C., Costadopoulos, N., Ali, S.: Hybrid secure and scalable electronic health record sharing in hybrid cloud. In: 5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud), pp. 185–190 (2017)
3. Arora, A., Khanna, A., Rastogi, A., Agarwal, A.: Cloud security ecosystem for data security and privacy. In: 7th International Conference on Cloud Computing, Data Science and Engineering, pp. 288–292 (2017)
4. Shrestha, M.N., Alsadoon, A., Prasad, C.P., Houran, L.: Enhanced eHealth framework for security and privacy in healthcare. In: 6th International Conference on Digital Information Processing and Communications (ICDIPC), pp. 75–79 (2016)
5. Suresh, S.: Highly secured cloud based personal health record model. In: International Conference on Green Engineering and Technologies (IC-GET), pp. 1–4 (2015)
6. Liu, Z., Weng, J., et al.: Cloud-based electronic health record system supporting fuzzy keyword search. Soft Comput. **20**(8), 3243–3255 (2016)
7. US Department of Health and Human Services: Health Insurance Portability and Accountability Act (2017). http://www.hhs.gov/ocr/privacy
8. Bahga, A., Madisetti, V.K.: Healthcare data integration and informatics in the cloud. Computer **48**(2), 50–57 (2015)
9. Fang, R., et al.: Computational health informatics in the big data age: a survey. ACM Comput. Surv. **49**(1), 12 (2016)
10. Zhou, J., Cao, Z., Dong, X., Lin, X.: PPDM: a privacy-preserving protocol for cloud-assisted e-healthcare systems. IEEE J. Sel. Top. Sig. Process. **9**(7), 1332–1344 (2015)
11. Zhou, J., et al.: PSMPA: patient self-controllable and multi-level privacy-preserving cooperative authentication in distributed m-healthcare cloud computing system. In: IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 6, pp. 1693–1703, 1 June 2015
12. Qureshi, B.: Towards a digital ecosystem for predictive healthcare analytics. In: 6th International Conference on Management of Emergent Digital EcoSystems (MEDES 2014), pp. 34–41 (2014)

13. Page, A., et al.: QT clock to improve detection of QT prolongation in long QT syndrome patients. Heart Rhythm **13**(1), 190–198 (2016)
14. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Trans. Inf. Theor. **22**(6), 644–654 (2006)
15. Qureshi, B., et al.: Performance of a low cost Hadoop cluster for image analysis in cloud robotics environment. Procedia Comput. Sci. **82**, 90–98 (2016)