International Journal of Electrical and Computer Engineering (IJECE)
Vol. 9, No. 5, October 2019, pp. 4258~4265
ISSN: 2088-8708, DOI: 10.11591/ijece.v9i5.pp4258-4265

Sensors data collection framework using mobile identification with secure data sharing model

Fulayjan Alanazi[1], Ahmed Elhadad[2], Safwat Hamad[3] and A Ghareeb[4]
[1,2]Department of Computer Science and information, Faculty of Science and Art, Jouf University, Saudi Arabia
[2,4]Department of Mathematics and Computer Science, Faculty of Science, South Valley University, Egypt
[3]Department of Scientific Computing, Faculty of Computer and Information Sciences, Ain Shams University, Egypt
[4]Department of Mathematics, Faculty of Science, Al-Baha University, Saudi Arabia

Article Info

Article history:

Received Feb 7, 2019
Revised Apr 4, 2019
Accepted Apr 25, 2019

Keywords:

Data sharing
Proxy
Re-encryption
Sensors

ABSTRACT

Sensors are the modules or electronic devices that are used to measure and get environmental events and send the captured data to other devices, usually computer processors allocated on the cloud. One of the most recent challenges is to protect and save the privacy issues of those sensors data on the cloud sharing. In this paper, sensors data collection framework is proposed using mobile identification and proxy re-encryption model for data sharing. The proposed framework includes: identity broker server, sensors managing and monitoring applications, messages queuing sever and data repository server. Finally, the experimental results show that the proposed proxy re-encryption model can work in real time.

Corresponding Author:

Fulayjan Alanazi,
Department of Computer Science and information,
Faculty of Science and Art, Aljouf University. Al Qurayyat, Saudi Arabia.
Email: fulayjan@ju.edu.sa

1. INTRODUCTION

Recently, Internet has become an indispensable communication service in our life since it facilitates the exchange of numerous types of digital data such as text, images, audio tracks, videos and 3D graphical objects. All those kinds of digital data are stored, transferred and shared among different users globally using cloud data sharing service [1-7]. Cloud data sharing, also called online data sharing, is a system in which users can store their data on a server over the Internet [8]. Furthermore, cloud data sharing allows users to import/export data between the web object storage and multiple devices [8].

Sensors are the modules and electronic devices that are used to measure and get data from environmental events [1]. The process of recording information from physical environmental sensors onto cyberworld is called sensory data collection framework [9, 10]. Furthermore, sensory data collection is one of the three main concepts that governs the connection between wireless sensor networks (WANs) and the Internet of Things (IoTs) [11, 12].

There are many proposed sensory data collection techniques [12]. One of those mathematical modeling techniques is proposed in [13], this method is based on a spatial-correlation where partial sensor data is collected and Multivariate Gaussian model estimates the non-transmitted data. The authors in [14] proposed a probabilistic technique called an energy efficient k-coverage algorithm which build a coverage network by expecting the minimum nodes. Recently, compressed sensory data collection techniques became common for data recording [15-17]. In general, the compression methods collect the data from the sensors in a time interval and sensors data are synchronized [12].

In this paper, sensors data collection framework proposed using mobile identification and proxy re-encryption model supports the sensory data sharing. The rest of this paper organized as the following: section II is the system overview, Section III is entities enrolment and communication schemes, section IV is the proxy re-encryption model for data sharing section V is results and dissection.

2. THE SYSTEM OVERVIEW

The proposed framework can be divided into four main parts: the first part stands for identity broker server which refers to identity and access management system; the second stands for the sensors managing and monitoring applications; the third part stands for the messages queuing sever which will be concerned mainly by the Advanced Message Queuing Protocol (AMQP). Finally, all data will be stored in bearer-only access data repository server. The contribution and relationship among the main parts of the framework in a simplified way is showed in Figure 1. In what follows we provide a more detailed description of the functional role played by each one of these architectural components.

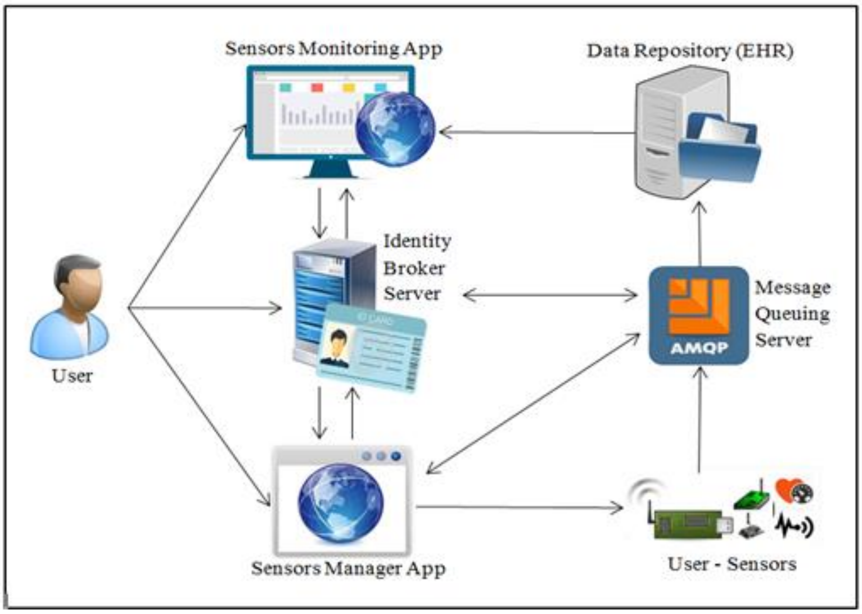


Figure 1. Overview diagram

2.1. Identity broker server

An identity broker server is a kind of an identity and access management (IdAM) solution which aims to provide a user-centric and centralized way for managing identities across different process domains. In addition, it was a middleware service to connect securely various framework service providers with users. The identity broker supports other features such as: Single-Sign on using mobile identification, Account Management, Standard Protocols and Authorization to create a trust relationship between the framework main parts. Moreover, an existing account can be linked with one or more identities from different identity providers or even created based on the identity information obtained from other providers such as Facebook, Google or Twitter.

2.2. Sensors manager and monitoring applications

The proposed framework is including two web applications: the Sensors Manager application will be used for creating a new sensor by adding the required information for the system to acquire the data from this sensor. The required information depends on the type of the sensor and the connection type while the system generates a unique ID for each sensor. In the same context, the sensor manager application enables the user to modify the sensor information in the case of parameters changing such as the connection type or data port. On the other hand, the application enables the user to delete the sensor information in the case of not using this sensor any more. However, The Sensor Monitoring application is the graphical user interface to display the various sensor data collection.

2.3. Message queuing server

The message Queuing server is a generally server using Advanced Message Queuing Protocol (AMQP) which aims to receive all the sensors data in asynchronous matter. Simply, the Message Queuing server consists of producer, Queue and consumer. The producer is the first station to receive the data from the various sensors then it will be transferred to the queue which finally delivered it to the consumer. The benefits of using the Message Queuing server architecture is to support receiving many data from different sensors in the same time. In addition, it allows get sensors data even the user offline.

2.4. Sensors manager and monitoring applications

Data Repository Server uses a web service application to provides all requests of data from the sensors monitoring application. It also stores the data collected from the consumer in the Message queuing server. The data in the repository server is secured from any unauthorized access. Only and only if the request with valid token can Get/Post data in the repository server.

3. ENTITIES ENROLMENT AND COMMUNICATION SCHEMES

In this framework, we will focus mainly on developing securely sensors data collection, monitoring and sharing with the allowance of the user. The proposed framework consists of registering a new user using mobile identification by creating a strong key; this key will be kept in the mobile for singing in procedure. In the same time, the user information details will be stored in the identity broker server with keeping user privacy during communication between framework applications.

Figure 2 illustrates registering a new user in the proposed framework. The registration process will be specified in the user, mobile and the identity broker server, where the process will be as the following:

– Registering: The user will fill the basic data such as First Name, Family Name or Email.
– Verifying: The identity broker will build a new session of user public key and encode it as a QR code.
– Confirming: The User will use the mobile QR code scanner to confirm his digital identity using this mobile phone.

After registration take place, the user can use the mobile phone as a digital identification; so, no user name or password is required to sign in. Consequently, the sensors management application will be used by the user to create, edit or delete sensors information as shown in Figure 3. For the first time of logging in the sensors manager application, the user is automatically redirected to the identity broker sever for getting the log in information from the mobile phone.

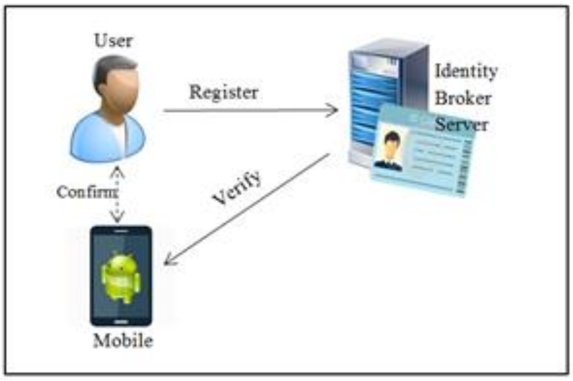


Figure 2. Register new user

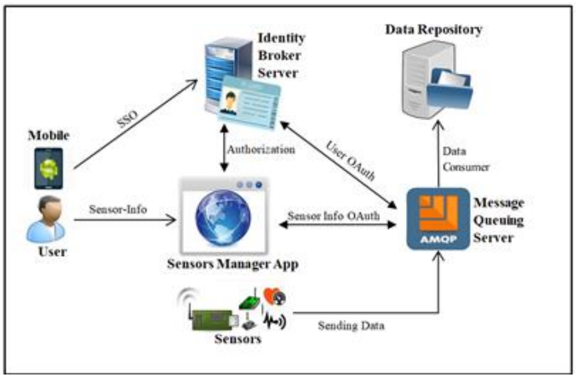


Figure 3. Sensor managing

The registered sensor, sending the data of the current state through the selected channel directly to the producer of the message queuing server. While the message queuing server will use the OAuth 2.0 protocol [18] to authorize and authenticate the sensor contact by sensors manager application and identity broker server respectively. After that, the message queuing server consumer delivers the data to the data repository server to be stored.

Figure 4 shows the sensors monitoring procedure. The sensors monitoring application requests user sign in using the mobile identification while the identity broker server authorizes a single sign on during the browsing process. The identity broker server provides the sensor monitoring application by a valid token to get the required data from the data repository server.

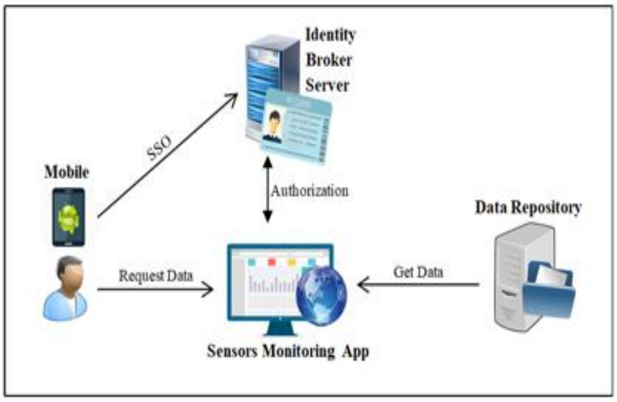


Figure 4. Sensors monitoring

The active novel scenario in the proposed framework is sharing sensors data authorization, which allows the owner of the data to share them with a requester who needs to benefit of this data. In this scenario, the requester requests a specific user sensor/s data using the sensors monitoring application which redirects the requester to the identity broker server for identification. After identifying the requester, the identity broker server send notification to the owner's mobile to request a permission for enabling the requester to get the sensor/s data. The owner grants or denies this request. In the case of granting the request, the owner can select which privileges can be used. Finally, the identity broker server authorizes the request and provides the sensors monitoring application with OAuth-token to get the required data from the data repository server. Figure 5 illustrates the entities diagram of the sharing data authorization scenario.

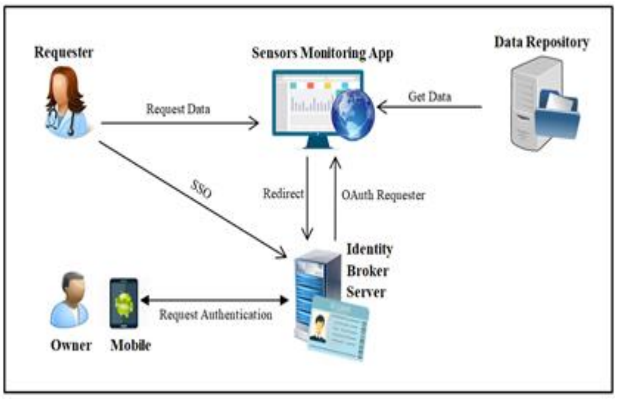


Figure 5. Sharing data authorizing

4. PROXY RE-ENCRYPTION MODEL FOR DATA SHARING

Hence, we propose a unidirectional proxy re-encryption model. The proposed model allows Alice to share encrypted plaintext on a public server with Bob using Bob's key without disclosing Alice's encryption key to Bob. Figure 6 shows an overview of the proposed model framework. In Alice's station, Alice creates the plaintext and encrypts it using *Key1*. Then, she sends the ciphertext to the public server via the proxy station. Therefore, Eve - the eavesdropper - cannot recover the plaintext. In Bob's station, Bob is authorized to recover Alice's plaintext, so he requests the ciphertext from the public server via the proxy station and the proxy station re-encrypts the ciphertext using the Rekey. Finally, Bob decrypts the Reciphertext using his own key(*Key2*). Hence, the proxy cannot decrypt the plaintext during the re-encryption process.

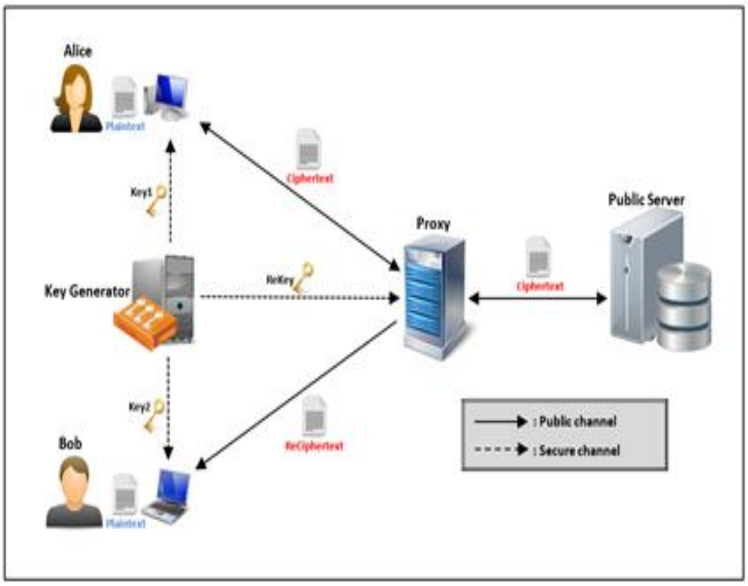


Figure 6. The proxy re-encryption model

The key generator station performs two different kind of procedures for generating the keys. The first procedure generates users' keys which are used by Alice and Bob. So, the following steps are used to generate any user key:

```
Algorithm 1: Key generation
Step1:  each key consists of unique 64 binary blocks each block includes 6-bits.
Step2:  each block divided into 3 parts each part is one of this set {00, 01, 10, 11}.
```

The second procedure generates the proxy Rekey which is depending on Key1 and Key2 of Alice and Bob keys respectively as the following:

```
Algorithm 2: Re-Key generation
Step1:   For each block in Key1:
         $\mathrm{ReKey} = \mathrm{block\ index(Key1)} - \mathrm{block\ index(Key2\ (block\ Key1))}$
Step2:   Convert integer indexes to binary using 8-bit representation.
```

So, Alice/Bob has a key of 384 binary bits and Rekey has 512 binary bits. The triple keys are transmitted securely to Alice, Bob and proxy stations. The Unidirectional cryptosystem technique include three procedures: Encryption, Decryption and Re-encryption. The Encryption procedure is some comprehensive steps of coding the plaintext and encrypts them using Alice's Key. In addition, it encloses the Ciphertext and Plaintext are the same length of string. The following steps describes the encryption procedure:

```
Algorithm 3: Encryption
Step1:  Read the Plaintext
        Step2: Convert the Plaintext to binary using 8-bit representation.
Step3:  Divide the binary Plaintext sequence into 6 bits blocks.
Step4:  For each block of the binary Plaintext calculate:
            $int\_index = index(\,Key_{blocks} \oplus bin\_plaintext_{block} = 0) - 1$
Step5:  Convert the int_index vector from integer to binary using 6-bit representation.
Step6:  Convert the binary sequence to string using 8-bit representation to get the Ciphertext.
```

In the same context, the decryption procedure is the inverse of the encryption process using the same key. The following steps describes the decryption procedure:

```
Algorithm 4: Decryption
Step1:  Read the Ciphertext
Step2:  Convert the Ciphertext to binary using 8-bit representation.
Step3:  Divide the binary Ciphertext sequence into 6 bits blocks.
Step4:  Convert each binary block into integer.
Step5:  For each integer value of the vector calculate:
        $bin\_seq = Key(int\_index + 1)$
Step6:  Convert the bin_seq from binary to string using 8-bit representation to get the Plaintext.
```

Finally, the Re-encryption procedure takes place when Bob is delegated by Alice to decrypt the Ciphertext. So, the proxy re-encrypts the Ciphertext using the corresponding Rekey and the following procedure:

```
Algorithm 5: Re-Encryption
Step1:  Read the Ciphertext.
Step2:  Convert the Ciphertext to binary using 8-bit representation.
Step3:  Divide the binary Ciphertext sequence into 6 bits blocks.
Step4:  Convert each binary block into integer.
Step5:  For each integer value of the vector calculate:
        $int\_ReCipher = int\_Cipher - ReKey(int\_Cipher + 1)$
Step6:  Convert the int_Recipher vector from integer to binary using 6-bit representation.
Step7:  Convert the binary sequence to string using 8-bit representation to get the
        ReCiphertext.
```

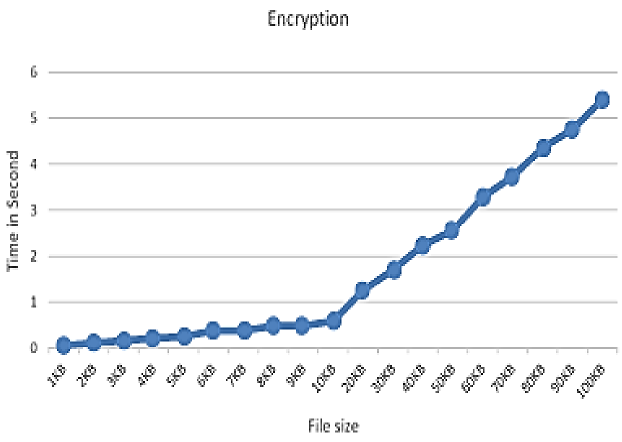
The unidirectional encryption proxy model is defined as a tuple E=(*KeyGen*, *Enc*, *Dec*, *ReEnc*) the key generation procedure *KeyGen* generates keys for Alice and Bob. Then, it generates one more key for the proxy. Alice encrypts the plaintexts using the *Enc* procedure and decrypts them using the *Dec* procedure. Whenever Bob wants to decrypt the ciphertext, He asks the proxy for help, the proxy uses the *ReEnc* procedure to transform the ciphertext into different ciphertext and sends it to Bob. Bob applies the Dec procedure to the received ciphertext and gets the original plaintext. The following equation reflects the way of the proposed unidirectional model work for the re-encryption process:

$$\begin{aligned} Ciphertext = Enc_{Key1}(Plaintext) \rightarrow Reciphertext = ReEnc_{Rekey}(Ciphertext) \\ \rightarrow Dec_{Key2}(ReCipher) = Plaintext \end{aligned}$$

5. RESULTS AND DISSECTION

The proposed framework was appointed a several open source libraries and applications. The Identity Broker Server was used a Keycloak server-Version: 3.0.0 [19]. The Message Queuing Server was used RabbitMQ server-Version: 3.6.9 [20]. The proposed proxy re-encryption model was implemented to evaluate the performance of the proposed method. Furthermore, the plaintext composed of randomly selected data to produce various file size starting from 1 KB till 100 KB. In addition, the simulation was implemented using Intel(R) Core (TM) i7-6700MQ CPU, 3.40 GHz, 64-bit windows 10 operating system with 16GB of RAM and MATLAB version: 9.0.0.341360 (R2016a).

Two seeds were used to generate random keys for Alice and Bob. The proxy Rekey was generated by using Alice and Bob keys. The execution time performances are 0.0003 sec and 0.0002 sec to generate Alice and Bob keys respectively. However, the re-encryption key generation execution time is 0.01 sec. Figure 7 shows the execution time of the proposed Encryption procedure for various plaintexts using Alice's key. As noted, the computed time increases significantly by increasing the file size. Thus, to encrypt a 1KB, 10KB and 100KB files take about 0.06, 0.59 and 5.41 seconds respectively. However, Figure 8 shows the proposed Re-Encryption procedure execution time cost for the corresponding plaintexts file size. As shown, the re-encryption process requires a little time execution while it take about 1.06, 4.65 and 45.70 milliseconds to encrypt a 1KB, 10KB and 100KB respectively which is compatible with proxy re-encryption.



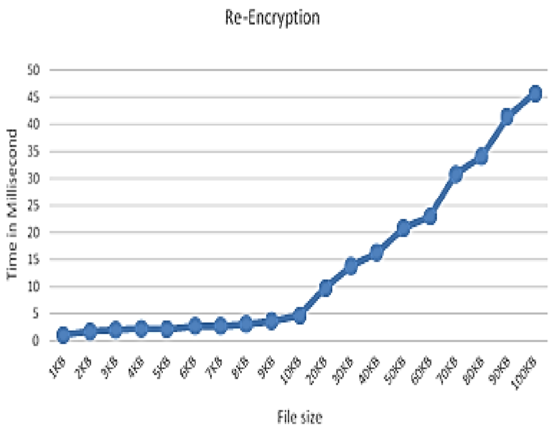


Figure 7. Encryptiom procedure execution time

Figure 8. Re-encryption procedure execution time

Hence, the Decryption procedure will be applied for both the *Cipher* and *ReCipher* using Alice and Bob Keys respectively. Figure 9 shows the decryption procedure execution time for Alice and Bob which are the same for the various files size. Thus, it will take 1.70, 9.62 and 93.63 milliseconds for Alice to decrypt a 1KB, 10KB and 100kB message files while it will take 0.96, 6.62 and 79.00 milliseconds for Bob.

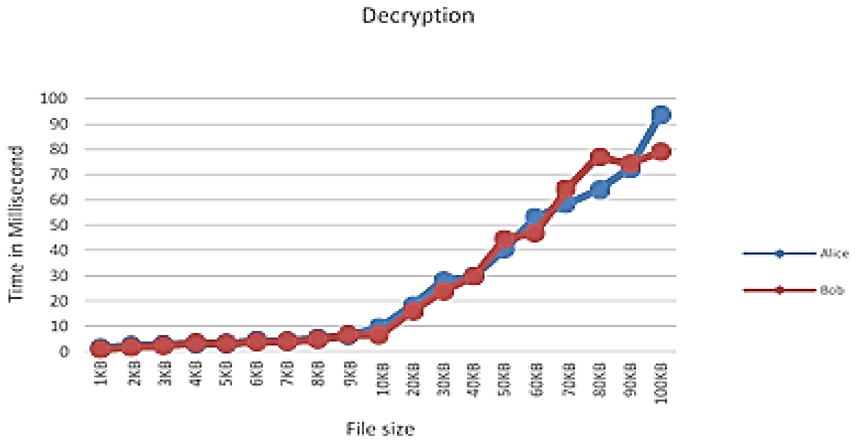


Figure 9. Decryptiom procedure execution time

6. CONCLUSION

In this paper, we propose sensors data collection framework using mobile identification. The proposed framework supports data sharing using a proxy re-encryption model. The execution time of the proposed model to encrypt a 1KB, 10KB and 100KB files take about 5.06, 0.59 and 5.41 seconds respectively. While, the re-encryption process requires a little time execution while it take about 1.06, 4.65 and 45.70 milliseconds to encrypt a 1KB, 10KB and 100KB respectively which is compatible with proxy re-encryption.

ACKNOWLEDGEMENTS

The second author would like to express his sincere thanks to professors Luís Antunes and Manuel Correia, CRACS & INESC TEC - Porto, Portugal for help and support.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, *et al.*, "A view of cloud computing," *Communications of the ACM,* vol. 53, pp. 50-58, 2010.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, *et al.*, "Above the clouds: A berkeley view of cloud computing," Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley 2009.

[3] A. Fu, S. Yu, Y. Zhang, H. Wang, and C. Huang, "NPP: A New Privacy-Aware Public Auditing Scheme for Cloud Data Sharing with Group Users," *IEEE Transactions on Big Data,* 2017.

[4] M. M. Hassan, K. Lin, X. Yue, and J. Wan, "A multimedia healthcare data sharing approach through cloud-based body area network," *Future Generation Computer Systems,* vol. 66, pp. 48-58, 2017.

[5] J. Shen, T. Zhou, D. He, Y. Zhang, X. Sun, and Y. Xiang, "Block design-based key agreement for group data sharing in cloud computing," *IEEE Transactions on Dependable and Secure Computing,* 2017.

[6] M. Sun, C. Ge, L. Fang, and J. Wang, "A proxy broadcast re-encryption for cloud data sharing," *Multimedia Tools and Applications,* pp. 1-15, 2017.

[7] J. Li, Y. Zhang, X. Chen, and Y. Xiang, "Secure attribute-based data sharing for resource-limited users in cloud computing," *Computers & Security,* vol. 72, pp. 1-12, 2018.

[8] T. Galibus, V. V. Krasnoproshin, R. de Oliveira Albuquerque, and E. P. de Freitas, *Elements of cloud storage security: concepts, designs and optimized practices*: Springer, 2016.

[9] C. Konstantopoulos, G. Pantziou, D. Gavalas, A. Mpitziopoulos, and B. Mamalis, "A rendezvous-based approach enabling energy-efficient sensory data collection with mobile sinks," *IEEE Transactions on Parallel and Distributed Systems,* vol. 23, pp. 809-817, 2012.

[10] P. Madhumathy and D. Sivakumar, "Enabling energy efficient sensory data collection using multiple mobile sink," *China Communications,* vol. 11, pp. 29-37, 2014.

[11] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials,* vol. 17, pp. 2347-2376, 2015.

[12] S. Cheng, Z. Cai, and J. Li, "Approximate Sensory Data Collection: A Survey," *Sensors,* vol. 17, p. 564, 2017.

[13] B. Gedik, L. Liu, and S. Y. Philip, "ASAP: An adaptive sampling approach to data collection in sensor networks," *IEEE Transactions on Parallel and distributed systems,* vol. 18, pp. 1766-1783, 2007.

[14] C. Li, Z. Sun, H. Wang, and H. Song, "A novel energy-efficient k-Coverage algorithm based on probability driven mechanism of wireless sensor networks," *International Journal of Distributed Sensor Networks,* vol. 12, pp. 1-9, 2016.

[15] S. Li, L. Da Xu, and X. Wang, "Compressed sensing signal and data acquisition in wireless sensor networks and internet of things," *IEEE Transactions on Industrial Informatics,* vol. 9, pp. 2177-2186, 2013.

[16] Y. Tang, B. Zhang, T. Jing, D. Wu, and X. Cheng, "Robust compressive data gathering in wireless sensor networks," *IEEE Transactions on Wireless Communications,* vol. 12, pp. 2754-2761, 2013.

[17] M. T. Nguyen and K. A. Teague, "Compressive sensing based random walk routing in wireless sensor networks," *Ad Hoc Networks,* vol. 54, pp. 99-110, 2017.

[18] D. Hardt, "The OAuth 2.0 authorization framework," 2012.

[19] *Keycloak*. Available: http://www.keycloak.org/index.html

[20] *RabbitMQ - Messaging that just works*. Available: https://www.rabbitmq.com/