NOVEL APPROACH TO SECURE PERSONAL HEALTH RECORDS DATA IN A SHARING ENVIRONMENT

[1]More Poonam Chandrasen, [2]Dr.B.M.Patil.

[1]M.Tech Student, [2]Department of PG,
[1]Department of Post Graduation (Computer Sciences and Information Technology),
[1]MBESs College of Engineering, Ambajogai, MS, India.

Abstract : cloud based services in the healthcare sector has resulted in cost effective and convenient exchange of Personal Health Records (PHRs) among several participating entities of the e-Health systems. Storing the confidential health information to cloud servers is not secure, there is issues such as revelation or theft of data and there is need for the development of methodologies that ensure the privacy of the PHRs.

Therefore, we use methodology called SeSPHR for secure sharing of the PHRs in the cloud. The SeSPHR scheme ensures patient-centric control on the PHRs and preserves the confidentiality of the PHRs. The patients store the encrypted PHRs on the un-trusted cloud servers and selectively grant access to different types of users on different portions of the PHRs. A semi-trusted proxy called Setup and Re-encryption Server (SRS) is introduced to set up the public/private key pairs and to produce the re-encryption keys. The methodology is secure against insider threats and also enforces a forward and backward access control.

Confidentiality of PHR records are maintained by providing password security. Analyze and verify working of methodology with help of SEED based XOR technique.

Index Terms - personal health record (PHR), Re-encryption Server (SRS), SeSPHR (secure sharing of personal health record)

1 INTRODUCTION

Healthcare system includes domains, such as patients, hospital staff including the doctors, nursing staff, pharmacies, and clinical laboratory personnel, insurance providers, and the service providers Generally, the PHRs contain information, such as: (a) demographic information, (b) patients' medical history including the diagnosis, allergies, past surgeries, and treatments, (c) laboratory reports, (d) data about health insurance claims, and (e) private notes of the patients about certain important observed health conditions. The PHRs are managed through the Internet based tools to permit patients to create and manage their health information as lifelong records that can be made available to those who need the access.

With the help of SeSPHR methodology communication and interaction between patients and PHR users. PHR users are pathologist, radiologist, doctors, pharmacist, friends and family. Storing the private health information to cloud servers managed by third parties is not secure .there is possibility of access of unauthorized person. There is major risk to store and maintain privacy of the PHRs stored in public clouds that are managed by commercial service providers. The PHRs are stored on the third-party cloud storage, they should be encrypted in such a way that neither the cloud server providers nor the unauthorized entities should be able to access the PHRs

A methodology called Secure Sharing of PHRs in the Cloud (SeSPHR) is used to administer the PHR access control mechanism managed by patients themselves. The methodology preserves the confidentiality of the PHRs by restricting the unauthorized users. Generally, there are two types of PHR users in the proposed approach, namely: (a) the patients or PHR owners and (b) the users of the PHRs other than the owners, such as doctors and physicians, health insurance companies' representatives, pharmacists, radiologist, pathologist and members or friends of patients. The patients as the owners of the PHRs are permitted to upload the encrypted PHRs on the cloud by selectively granting the access to users over different portions of the PHRs.

The levels of access granted to various categories of users are defined in the Access Control List (ACL) by the PHR owner. For example, the family members or friends of the patients may be given full access over the PHRs by the owner. Similarly, the representatives of the insurance company may only be able to access the portions of PHRs containing information about the health insurance claims while the other confidential medical information, such as medical history of the patient is restricted for such users.

SRS set the public/private key pairs and producing the decryption keys for the authorized users only this avoids overhead at end users. The methodology considers the cloud servers as the untrusted entity and therefore, introduces a semi-trusted server called the Setup and Re-encryption Server (SRS) as the proxy. Proxy Re-encryption based approach is used for the SRS to generate the re-encryption keys for secure sharing of PHRs among the users. The PHRs are encrypted by the patients or PHR owners and only the authorized users having the keys issued by the SRS can decrypt the PHRs.

This methodology is more secure because the users are granted access to the specific portions of PHRs.

2. LITERATURE REVIEW

1. M. H. Au, T. H. Yuen, J. K. Liu, W. Susilo, X. Huang, Y. Xiang, and Z. L. Jiang proposed general framework for secure sharing of personal health records in cloud system. This system enables patients to securely store and share their PHR in cloud server .for research proposes the doctor can refers the patient's medical record, where they are required.[1]
2. IIoT (Industrial internet of the things) technique for health monitoring is proposed by M. Shamim Hossain, Ghulam Muhammad, et al.This technique is useful for interconnection between medical devices and also useful for elderly and disabled people. Health IIoT is combination of communication technologies, interconnected apps,devices ,sensor and people which act as smart system to monitor, track and store patient's information.[12]
3. PHR is electronically store, manage and share their personal health records proposed by J. Li. [4] this papers analyzes privacy and security issues with PHR.This paper includes high-minded privacy principles such as security audits, management, and independent privacy. This principles used in the web-based PHR system.
4. With the help of attributes-based encryption scalable and secure of personal health records in cloud computing is proposed by M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou. ABE (Attribute-based encryption) technique is uses to achieve fine-grained, scalable data access control to PHR and to encrypt patient's PHR file [3].
5. In encryption process, to achieve accurately accessing to medical records and corresponding to flexibility and efficiency, a new PHR access control scheme is proposed by T. S. Chen, C. H. Liu, T. L. Chen, C. S. Chen, J. G. Bau, and T.C. Lin in Secure Dynamic access control scheme of PHR in cloud computing [5]. With Lagrange interpolation polynomial to establish a secure and effective PHR information access scheme, it allows to accurately access to PHR with security and is suitable for multi-users. This scheme also dynamically supports multi-users in Cloud computing environments with personal privacy and offers legal authorities to access to PHR.

3. Proposed System

In this proposed systems provides confidentiality and privacy of patient's sensitive health record. The methodology used in proposed system is SeSPHR i.e. secure sharing of personal health records in the cloud. This methodology preserve the confidentiality of the restricting the unauthorized users. PHR are of two types the patients i.e. Owner and PHR user such as doctor, pathologist, farmacist, family and friends, radiologist, health insurance companies.

SRS i.e. semi trusted proxy is deployed to ensure the access control and generate the re-encryption keys for different users. Seed based encryption is used for secure sharing of personal health records. In this project there is access to multiple users but user who have grant permission is allowed to read the PHR of patient's.

3.1 Methodology of proposed model:

Here we are developing a web based application which will help users to login/authentication, upload and download medical records. The system will connect to one console application SRS, to generate the keys. The SRS also takes cares of permission per files and all data is kept in MySQL database. The internal communication between SRS and web-application is based on HTTP protocol. Proxy Re-encryption based approach is used for the SRS to generate the re-encryption keys for secure sharing of PHRs among the users. The PHRs are encrypted by the patients or PHR owners and only the authorized users having the keys issued by the SRS can decrypt the PHRs. Moreover, the users are granted access to the specific portions of PHRs as deemed important by the PHR owner.

SEED based XOR technique is used for encryption of files.

The simple XOR cipher is a type of additive cipher, an encryption algorithm that operates according to the principles:

$$A \oplus 0 = A,$$

$$A \oplus A = 0,$$

$$(A \oplus B) \oplus C = A \oplus (B \oplus C),$$

$$(B \oplus A) \oplus A = B \oplus 0 = B,$$

For example, the string "Wiki" (01010111 01101001 01101011 01101001 in 8-bit ASCII) can be encrypted with the repeating key 11110011 as follows:

```
     01010111 01101001 01101011 01101001
     11110011 11110011 11110011 11110011
   ----------------------------------------
=    10100100 10011010 10011000 10011010
```

And conversely, for decryption:

```
     10100100 10011010 10011000 10011010
     11110011 11110011 11110011 11110011
   ----------------------------------------
=    01010111 01101001 01101011 01101001
```

The XOR operator is extremely common as a component in more complex ciphers. By itself, using a constant repeating key, a simple XOR cipher can trivially be broken using frequency analysis. If the content of any message can be guessed or otherwise known then the key can be revealed. Its primary merit is that it is simple to implement, and that the XOR operation is computationally inexpensive. A simple repeating XOR (i.e. using the same key for xor operation on the whole data) cipher is therefore sometimes used for hiding information in cases where no particular security is required. The XOR cipher is often used in computer malware to make reverse engineering more difficult

3.2 safety and security requirements:

Use of hash function, SHA256, to create hash of password.

The SHA (Secure Hash Algorithm) is one of a number of cryptographic hash functions. A cryptographic hash is like a signature for a text or a data file. SHA-256 algorithm generates an almost-unique, fixed size 256-bit (32-byte) hash. Hash is a one way function – it cannot be decrypted back. This makes it suitable for password validation, challenge hash authentication, anti-tamper, digital signatures. SHA-256 is one of the successor hash functions to SHA-1, and is one of the strongest hash functions available.

3.3 Implementation of proposed model:

As shown in figure 1 as part of project implementation we have created 2 different modules namely key generation and Storage Server. The key generation modules takes responsibility to generating unique public/private keys for each file uploaded by patient/user into system. That key is then converted to hash number using SHA256 hashing technique. That hashed number is then used as final key to encrypt the file data which is just uploaded by user. On the other hand the server application maintains the database and the storage with additional logic of authentication and access management. Both the module are build using Java platform however key generation server act as client is no-gui application and storage server is web based application. Key Generation client developed using socket programming technique where it connects to server application to handle the incoming request. Storage Server is build using servlet and JSP as front end and MySql as back-end.

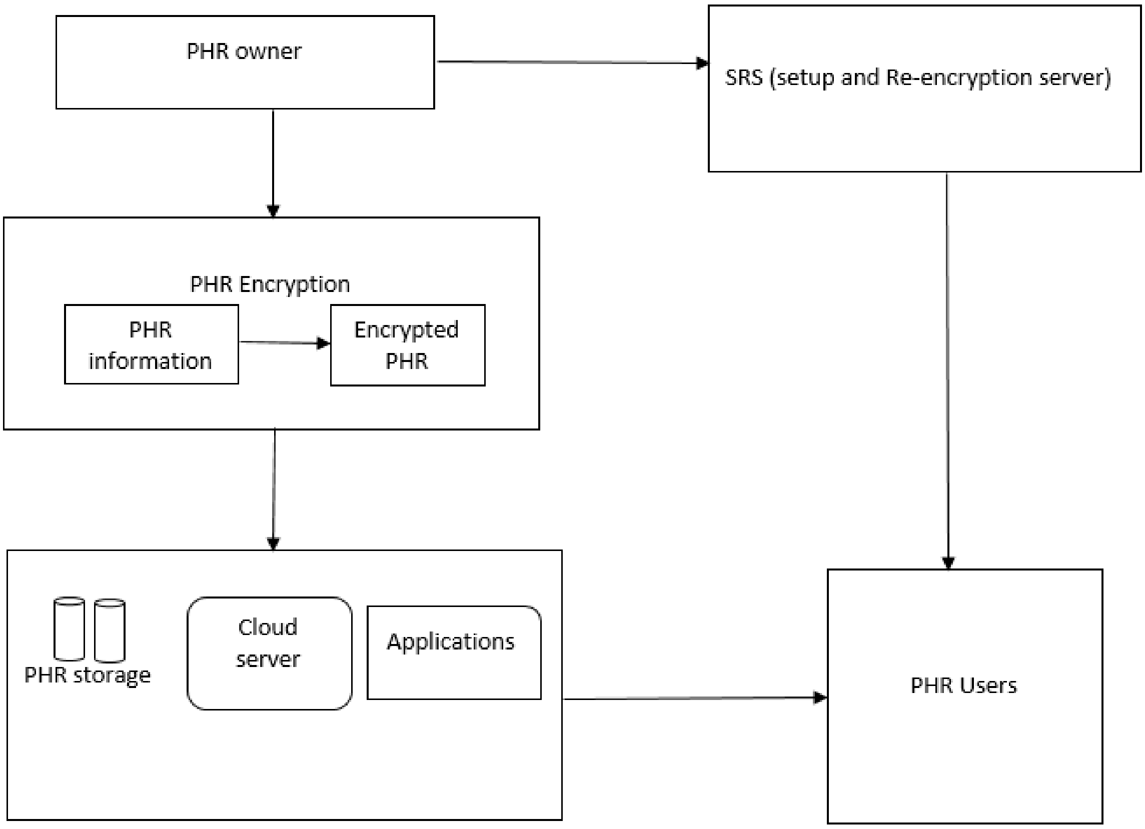

Figure 1: Architecture of the proposed model

We have provided following features into our system:

- User registration and authentication
- User can upload PHR record
- System can apply the algorithm to find frauds.

This project suggests and justifies that theses sharing technology has been beneficial for sharing the PHR record to users

Here we need to create 2 users role

1. Patient
2. PHR users

Steps for implementation:

- System should allow user to login
- System should verify the users.
- System should allow patient user to upload files
- System should allow patient user to give access to PHRs users
- System should allow patient to view uploaded files.
- System should allow PHR user to register
- System should allow PHR user to login
- System should verify the PHR users
- System should allow PHR user to download the file
- System should send the key for download the file

Working of methodology:

1. Patient upload the encrypted PHR
2. The client application for the PHR also generates the re-encryption parameters that are subsequently transmitted to the SRS.
3. If a user wants to access any portion of the PHR, only after authentication the user downloads the PHR from the cloud

4. User needs to obtain the corresponding decryption parameters from the SRS to decrypt PHR
5. The SRS check for requesting user and determines whether the access to the partition for which the user has requested the decryption parameters is granted by the PHR owner or not
6. According to the access permissions specified .the SRS will generate the corresponding parameters and will send those to the requesting user. The setup, key generation, and re-encryption phases are carried out at SRS.

4. RESULT

The performance of the system to securely share the PHRs among different types of users was evaluated by developing a client application in Java. Due to the fixed size of the prime number, the encryption and decryption process was carried out in the chunks of 64 bytes. The experiments were conducted on the computer having Intel® Core i5-2600 CPU @ 2.00 GHz with 4 GB memory.

The performance of the system to securely share the PHRs among different types of users was evaluated by developing a client application in Java. Due to the fixed size of the prime number, the encryption and decryption process was carried out in the chunks of 64 bytes. The experiments were conducted on the computer having Intel® Core i5-2600 CPU @ 2.00 GHz with 4 GB memory.

The performance of the proposed model is evaluated regarding generation, encryption, and decryption.

4.1 Key Generation

Setup and Re-encryption Server (SRS) generate the private/public key pairs for the users belonging to the set of authorized users. Performance of the system is depend on the key generation time for the systems with large numbers of users.it affects overall performance of the system. As shown in figure 4.1 key generation time is increased when the number of users increases. This figure shows the difference between key generating time between proposed model and previous model.

The time consumption for generating keys for 10, 100, 500, 1000, 5000, and 10,000 users in presented as shown in. Figure 2.For example time required for previous model to generate keys is 0.7 ms .key generation time decrease for proposed model i.e.0.5 ms. likewise key generation time required for proposed model with 10000-12000 users. Whereas time required for key generation for previous model required 0.7 ms.

The encrypted data stored by the PHR owners on the cloud and only the authorized users possessing valid re-encryption keys issued by a semi trusted proxy are able to decrypt the PHRs.Semi-trusted proxy play important role to generate and store the public/private key pairs for the users in the system. To preserving the confidentiality and ensuring patient-centric access control over the PHRs, the proposed system also administers the forward and backward access control for departing and the newly joining users, respectively.

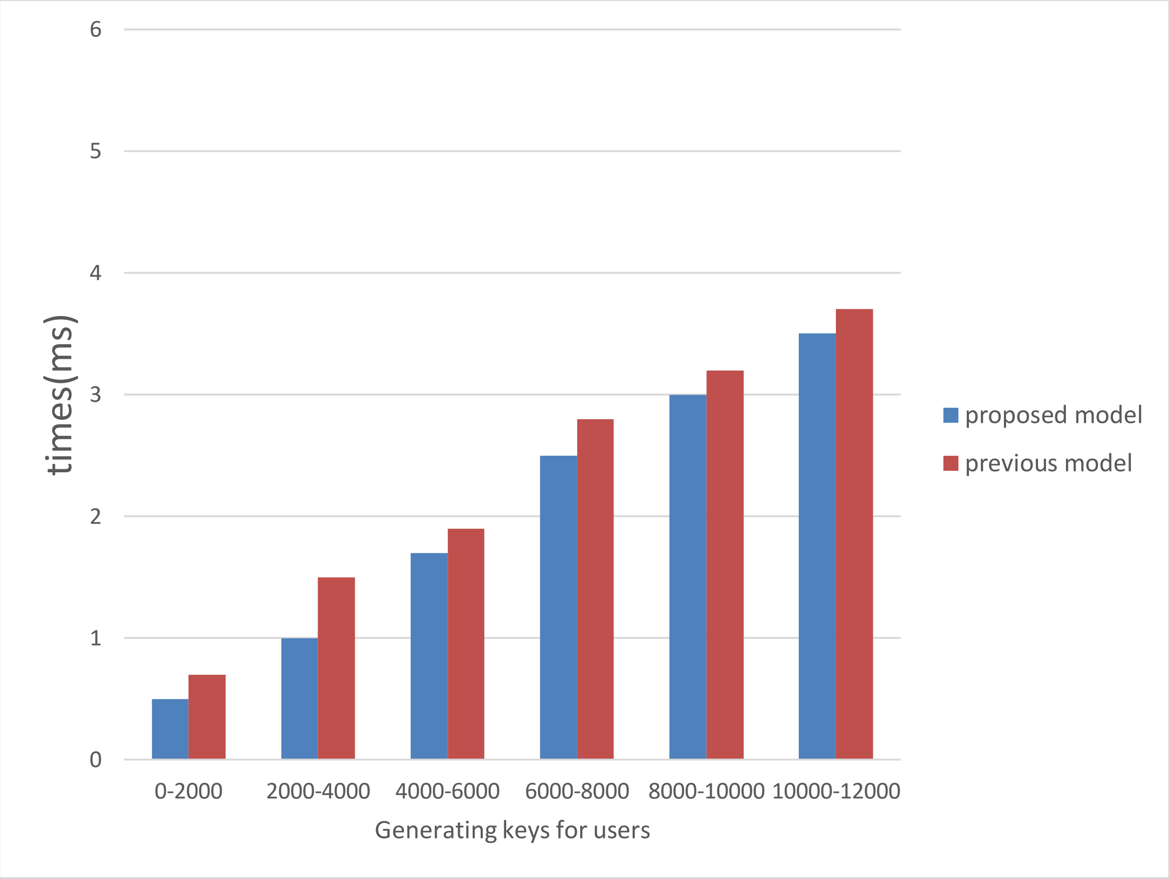

Figure 2: Time consumption for generating keys

4.2. The time consumption for Encryption and decryption

The time consumption of to encrypt and decrypt the data files of is also evaluated. The file sizes used for the experimentation are 50 KB, 100 KB, 200 KB, 500 KB, 800 KB, 1024 KB, 1500 KB, and 2048 KB. Figure 3 and figure 4 shows the time consumption for both the encryption and decryption operations for the files respectively. From Figure 3 we can see that with the increase in PHR file size, the encryption time also increases. Figure 3 shows difference between times required for encryption of PHR file of proposed model and previous model. For example, the encryption time for the file of size 50 KB for proposed model required 0.5 ms whereas the encryption time for previous model required 0.9 ms.

The time required for decryption of the PHR file of proposed model is less than the encryption time required for previous model as shown in figure 4

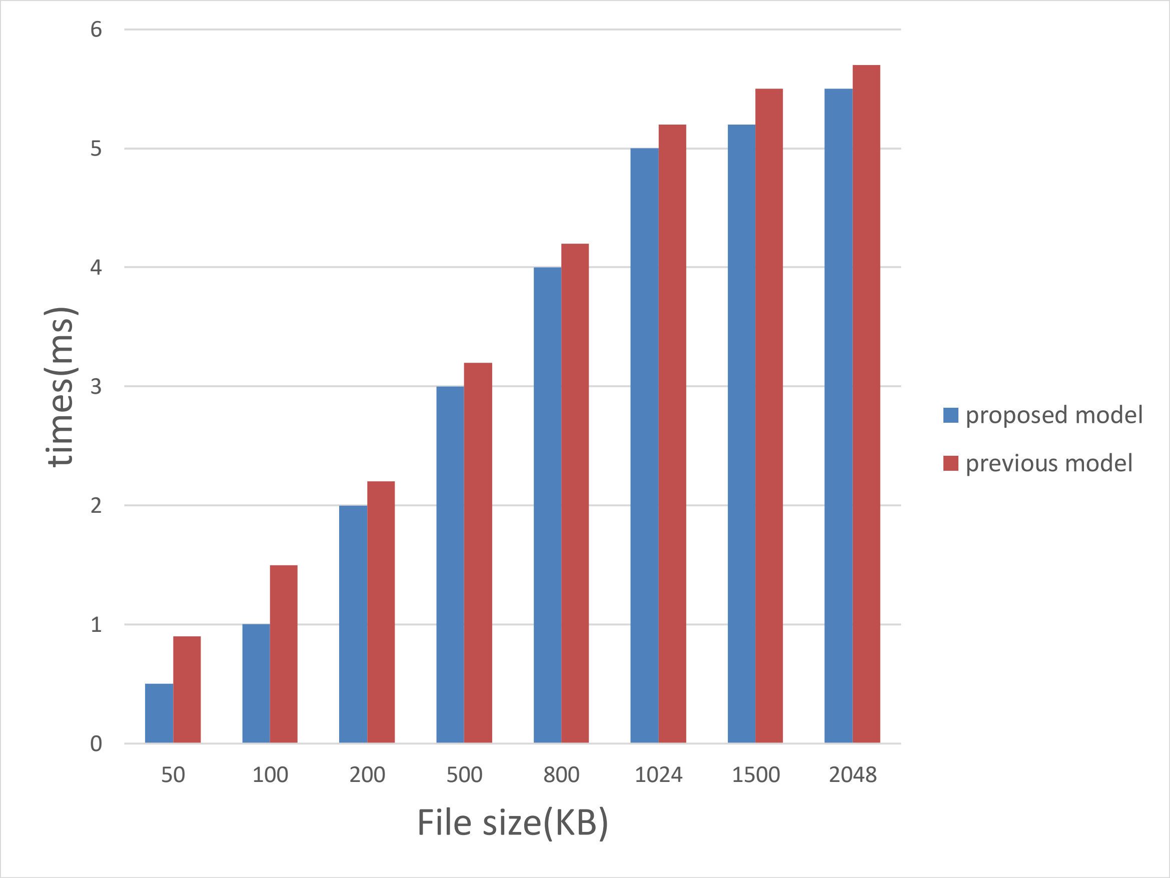

Figure 3: The time consumption for Encryption

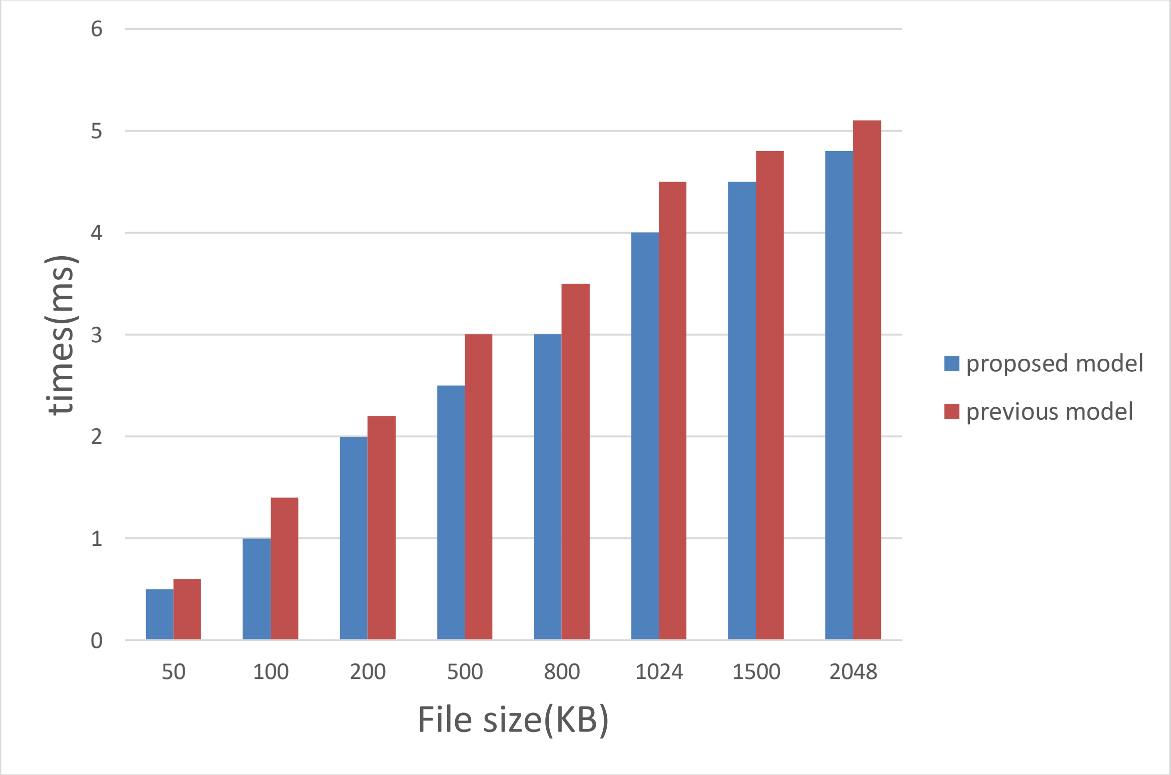

Figure 4: The time consumption for Decryption

Conclusion:

We implemented a fine-grained access control method in such a way that even the valid system users cannot access those portions of the PHR for which they are not authorized. The PHR owners store the encrypted data on the cloud and only the authorized users possessing valid re-encryption keys issued by a semi-trusted proxy are able to decrypt the PHRs. The experimental results exhibit the viability of the SeSPHR methodology to securely share the PHRs in the cloud environment.

References:

[1] M. H. Au, T. H. Yuen, J. K. Liu, W. Susilo, X. Huang, Y. Xiang, and Z. L. Jiang, “A general framework for secure sharing of personal health records in cloud system,” Journal of Computer and System Sciences, vol. 90, pp, 46-62, 2017

[2] A.Abbas, K. Bilal, L. Zhang, and S. U. Khan, “A cloud based health insurance plan recommendation system: A user centered approach, “Future Generation Computer Systems, vols. 4344, pp. 99-109, 2015.

[3] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, “Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption,” IEEE Transactions on Parallel and Distributed Systems, 2013, vol. 24, no. 1, pp. 131–143.

[4] J. Li, “Electronic personal health records and the question of privacy,” Computers, 2013, DOI: 10.1109/MC.2013.225.

[5] T. S. Chen, C. H. Liu, T. L. Chen, C. S. Chen, J. G. Bau, and T.C. Lin, “Secure Dynamic access control scheme of PHR in cloud computing,” Journal of Medical Systems, vol. 36, no. 6, pp. 4005– 4020, 2012.

[6] Z. Xiao and Y. Xiao, “Security and privacy in cloud computing,” IEEE Communications Surveys and Tutorials, vol. 15, no. 2, pp. 1–17, Jul. 2012.

[7] R. Wu, G.-J. Ahn, and H. Hu, “Secure sharing of electronic health records in clouds,” In 8th IEEE International Conference on Collaborative Computing: Networking, Applications and Work sharing (CollaborateCom), 2012, pp. 711-718

[8] A. N. Khan, ML M. Kiah, S. A. Madani, M. Ali, and S. Shamshirband, “Incremental proxy re-encryption scheme for mobile cloud computing environment,” The Journal of Supercomputing, Vol. 68, No. 2, 2014, pp. 624-651.

[9] D. C. Kaelber, A. K. Jha, D. Johnston, B. Middleton, and D. W. Bates, “A research agenda for personal health records (PHRs),” Journal of the American Medical Informatics Association, vol. 15, no. 6, 2008, pp. 729-736.

[10] S. Yu, C. Wang, K. Ren, and W. Lou, “Achieving secure, scalable and fine-grained data access control in cloud computing,” in Proceedings of the IEEE INFOCOM, March 2010, pp. 1-9

[11] A. Abbas and S. U. Khan, “A Review on the State-of-the-Art Privacy Preserving Approaches in E-Health Clouds,” IEEE Journal of Biomedical and Health Informatics, vol. 18, no. 4, pp. 1431-1441, 2014.

[12] M. Shamim Hossain, Ghulam Muhammad, et al. "Cloud-assisted Industrial Internet of Things (IIoT) –Enabled framework for health monitoring." 2016 evier B.V..