SeSPHR A Methodology for Secure Sharing of Personal Health Records in the Cloud

ABSTRACT

The widespread acceptance of 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. Nevertheless, storing the confidential health information to cloud servers is susceptible to revelation or theft and calls for the development of methodologies that ensure the privacy of the PHRs. Therefore, we propose a 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. Moreover, the methodology is secure against insider threats and also enforces a forward and backward access control. Furthermore, we formally analyze and verify the working of SeSPHR methodology through the High Level Petri Nets (HLPN). Performance evaluation regarding time consumption indicates that the SeSPHR methodology has potential to be employed for securely sharing the PHRs in the cloud.

**EXISTING SYSTEM**

* Chen *et al*. [12] introduced a method to exercise the access control dynamically on the PHRs in the multi-user cloud environment through the Lagrange Multiplier using the SKE. Automatic user revocation is the key characteristics of the approach. To overcome the complexities of the key management, a partial order relationship among the users is maintained. However, the scheme requires the PHR owners to be online when the access is to be granted or revoked. Contrary to the scheme presented in [12], our approach does not require the PHR owners to be online to grant the access over PHRs. In- stead the semi-trusted authority determines the access privileges for users and after successful authorization, calculates the re-encryption keys for the users requesting the access.
* The authors in [29] used a Digital Right Management (DRM) based approach to offer patient-centric access con- trol. The authors employed the Content Key Encryption (CKE) for encryption and the users with the lawful li- cense are permitted to access the health-data. First proxy re-encryption methodology was proposed in [33].
* The policy in [33] is based on ciphertext and the size of the ciphertext increases linearly with multi-use use whereas our policy of our technique is based on keys and it doesn’t affect the size of the ciphertext. This is due to the fact that the [33] requires the re-encryption step that is lacking in our methodology. An approach to securely share the PHRs in multi-owner setting, which is divided into diverse domains using the Attribute Based Encryp- tion (ABE) is presented by Li *et al.* [14]. The proposed methodology is based on the methodology originally pre- sented in [33].
* The approach uses proxy re-encryption technique to re-encrypt the PHRs after the revocation of certain user(s). In the approach, the intricacies and cost of key management have been effectively minimized and the phenomenon of on-demand user revocation has been improved. Despite its scalability, the approach is unable to efficiently handle the circumstances that require grant- ing the access rights on the basis of users’ identities.
* Xhafa *et al*. [30] also used Ciphertext Policy ABE (CP- ABE) to ensure the user accountability. Besides protecting the privacy of the users, the proposed approach is also capable of identifying the users that malfunction and dis- tribute the decryption keys to other users illegitimately.
* **Disadvantages**
* The system doesn’t implemented El-Gamal encryption which is effective to secure data.
* There is no dynamic data integrity proof instead manual.

**PROPOSED SYSTEM**

* The proposed system presents a methodology called Secure Sharing of PHRs in the Cloud (SeSPHR) 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 the family members or friends of patients, doctors and physicians, health insurance companies’ representatives, pharmacists, and researchers.
* 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. Each member of the group of users of later type is granted access to the PHRs by the PHR owners to a certain level depending upon the role of the user. The levels of access granted to various categories of users are de- fined 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.

**Advantages**

* The system presents a methodology called SeSPHR that permits patients to administer the sharing of their own PHRs in the cloud.
* The SeSPHR methodology employs the El-Gamal encryption and proxy re-encryption to ensure the PHR confidentiality.
* The methodology allows the PHR owners to selectively grant access to users over the portions of PHRs based on the access level specified in the ACL for different groups of users.

**SYSTEM REQUIREMENTS**

➢ **H/W System Configuration:-**

➢ Processor - Pentium –IV

➢ RAM - 4 GB (min)

➢ Hard Disk - 20 GB

➢ Key Board - Standard Windows Keyboard

➢ Mouse - Two or Three Button Mouse

➢ Monitor - SVGA

**Software Requirements:**

* Operating System - Windows XP
* Coding Language - Java/J2EE(JSP,Servlet)
* Front End - J2EE
* Back End - MySQL