**A CLOUD-BASED METHODOLOGY FOR SECURE PERSONAL HEALTH RECORDS SHARING**

**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.

|  |  |
| --- | --- |
| **EXSISTING SYSTEM** | **PROPOSED SYSTEM** |
| * The patients encrypt the PHRs by the patients through the public key of the Cloud Service Provider (CSP) and the CSP decrypts the record using the private key. * It can stores the health record and the location of the file (index), and subsequently encrypts them through the symmetric key encryption. * A semi-trusted authority called the SRS that re-encrypts the cipher text generated by the PHR owner and issues keys to the users that request access to the PHRs. | * 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. |
| **EXISTING ALGORITHM**  High Level Petri Nets (HLPNs) | **PROPOSED ALGORITHM:-**  A semi-trusted proxy called Setup and Re-encryption Server (SRS) |
| **ALGORITHM DEFINITION:-**  The petri nets are the tools that are employed to graphically and mathematically model the systems [20]. The petri nets can modela variety of systems that can be characterized as the parallel, concurrent, distributed, non-deterministic, asynchronous, and stochastic [21]. To model the working of the SeSPHR methodology, we used the HLPN, which is a variation of the traditional petri nets. The HLPN is a structure comprising of 7. | **ALGORITHM DEFINITION:-**  We present 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. |
| **DRAWBACKS:-**   * It has a low cost effective. * posed framework is never transmitted the PHR data. In-stead, the responsibility of the SRS is to manage the keys. | **ADVANTAGES:-**   * Despite the advantages of scalable, * agile, * cost effective, and * ubiquitous services offered by the cloud. |

**System model with a cloud server, a data owner and**

**data users.**

****

**MINIMUMSYSTEM REQUIREMENTS**

**HARDWARE REQUIREMENTS**

* PROCESSOR : DUAL CORE 2 DUO.
* RAM : 2GB DD RAM
* HARD DISK : 250 GB

**SOFTWARE REQUIREMENTS**

* FRONT END : J2EE (JSP, SERVLET)
* BACK END : MY SQL 5.5
* OPERATING SYSTEM : WINDOWS 7
* IDE : ECLIPSE