IOSR Journal of Engineering (IOSR JEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
PP 01-04

An Energy Efficient Multifactor Authentication Scheme For eHealth Meets Security

Atchaya ,.P[1] , Dr. PriyaRadhikadevi .T[2]
[1](PG Scholar, Dept of computer science and Engineering, Mailam Engineering College.)
[2](.Head of department ,Dept of computer science and Engineering, Mailam EngineeringCollege)

Abstract: *In the EXISTING SYSTEM, Personal health record (PHR) is an emerging patient-centric in Cloud Computing Servers. However, there is no Security in keeping privacy concerns of the Patient & could be exposed to those third party servers and to unauthorized parties. In the PROPOSED MODEL, a novel patient-centric framework and a suite of mechanisms for data access control to PHRs stored in semi-trusted servers. We leverage attribute based encryption (ABE) techniques to encrypt each patient's PHR file. Our scheme also enables dynamic modification of access policies or file attributes, supports efficient on-demand user/attribute revocation and break-glass access under emergency scenarios.*

I. Introduction

DOMAIN SPECIFICATION: Cloud computing takes the technology, services, and applications that are similar to those on the Internet and turns them into a self-service utility. The use of the word "cloud" makes reference to the two essential concepts:

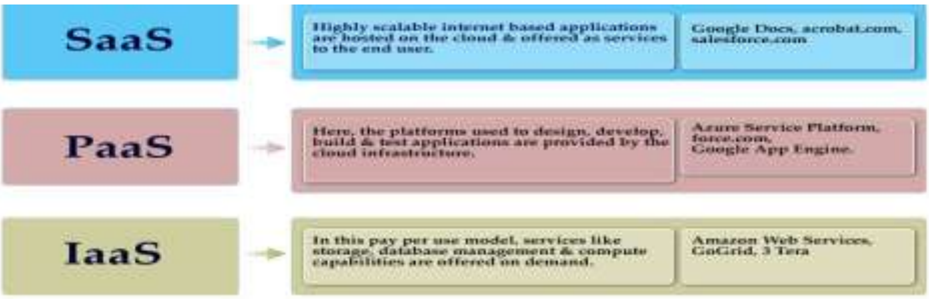
Abstraction: Cloud computing abstracts the details of system implementation from users and developers. Applications run on physical systems that aren't specified, data is stored in locations that are unknown, administration of systems is outsourced to others, and access by users is ubiquitous. by pooling and sharing resources. Systems and storage can be provisioned as needed from a centralized infrastructure, costs are assessed on a metered basis, multi-tenancy is enabled, and resources are scalable with agility.

In recent years, personal health record (PHR) has emerged as a patient-centric model of health information exchange. A PHR service allows a patient to create, manage, and control her personal health data in one place through the web, which has made the storage, retrieval, and sharing of the the medical information more efficient. Especially, each patient is promised the full control of her medical records and can share her health data with a wide range of users, including healthcare providers, family members or friends. Due to the high cost of building and maintaining specialized data centers, many PHR services are outsourced to or provided by third-party service providers, for example, Microsoft Health Vault.

Software As A Service (Saas): In this model, a complete application is offered to the customer, as a service on demand. A single instance of the service runs on the cloud & multiple end users are serviced.

Platform as a Service (Paas): Here, a layer of software, or development environment is encapsulated & offered as a service, upon which other higher levels of service can be built.

Infrastructure as a Service (Iaas): IaaS provides basic storage and computing capabilities as standardized services over the network. Servers, storage systems, networking equipment, data centre space etc.



SaaS	Highly scalable internet based applications are hosted on the cloud & offered as services to the end user.	Google Docs, acrobat.com, salesforce.com
PaaS	Here, the platforms used to design, develop, build & test applications are provided by the cloud infrastructure.	Azure Service Platform, force.com, Google App Engine.
IaaS	In this pay per use model, services like storage, database management & compute capabilities are offered on demand.	Amazon Web Services, GoGrid, 3 Tera

Fig:1.1 DOMAIN SPECIFICATION

II. Proposed System:

In the PROPOSED MODEL, a novel patient-centric framework and a suite of mechanisms for data access control to PHRs stored in semi-trusted servers. We leverage attribute based encryption (ABE) techniques to encrypt each patient's PHR file. Our scheme also enables dynamic modification of access policies or file attributes, supports efficient on-demand user/attribute revocation and break-glass access under emergency scenarios.

Advantages Of The Proposed System:

- Security level will be increased.
- Trustworthiness will also be maintained properly.
- The Java compiler does not produce an executable file, so Java programs can not execute under the operating system of your machine. Instead they execute inside a Java Virtual Machine, which is invoked using the java program of the JDK.
- Executing a Class File
- To execute a Java program the Java Developer's Kit provides a program called java. When executing that program with your class file as parameter the following happens:
- the Java Virtual Machine (JVM) is created inside your computer
- the JVM locates and reads your class files
- the JVM inspects your class file for any security violations
- the JVM executes, or interprets, your class file according to its instructions if possible
- Under Windows and Unix, execute a program by typing at the command prompt java Name, where Name is the name of the program (no extension). On a Macintosh, double-click the java icon and select the appropriate class file.

Uml Diagrams

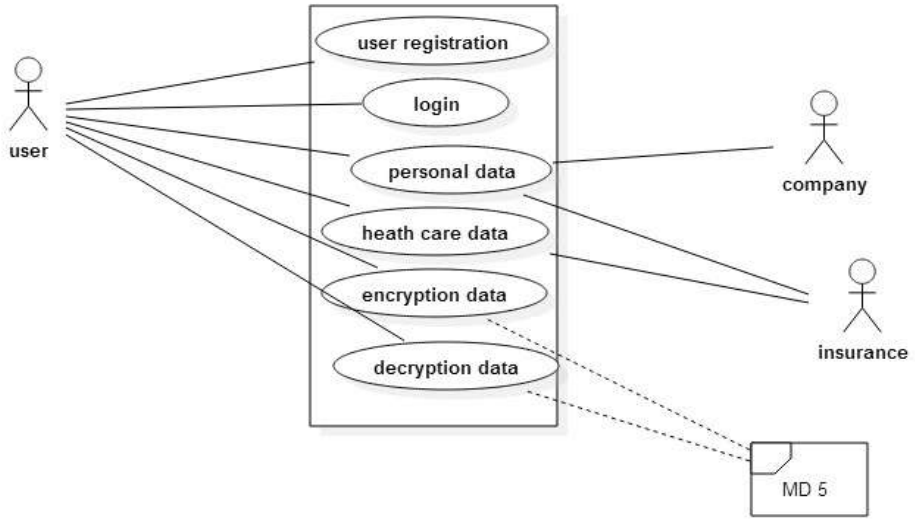
The standard is managed, and was created by, the Object Management Group. The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

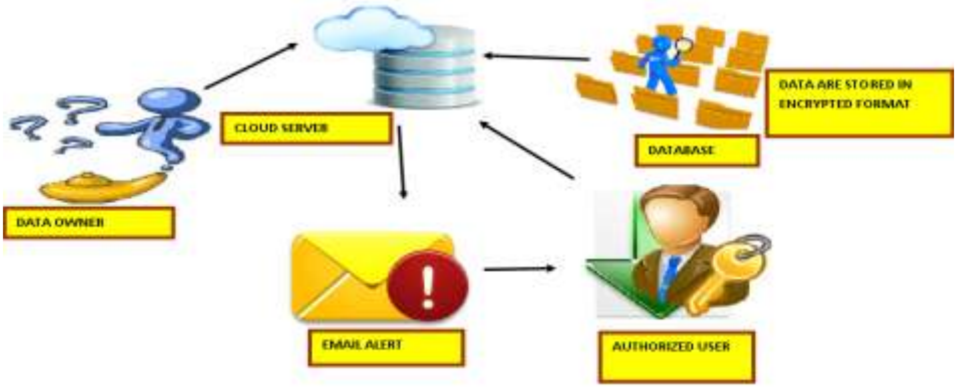
The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Be independent of particular programming languages and development process.
3. Provide a formal basis for understanding the modeling language.
4. Encourage the growth of OO tools market.
5. Support higher level development concepts such as collaborations, frameworks, patterns and components.

ALGORITHM/METHODOLOGY: Attribute based encryption(ABE)
REALTIMEAPPLICATION: Hospital management
CLASS DIAGRAM:

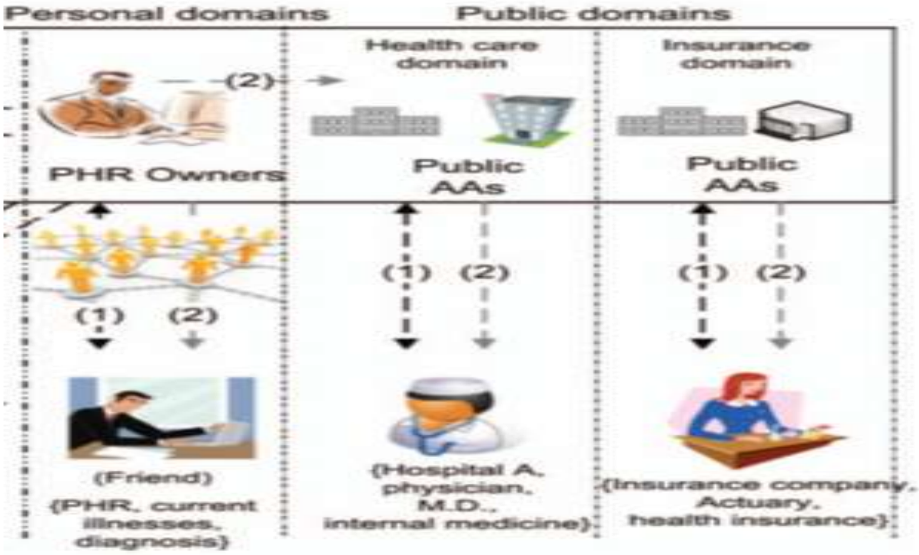


ARCHITECTURE DIAGRAM



.

ARCHITECTURE:


Personal domains
Public domains
Health care domain
Insurance domain
(2)
PHR Owners
Public AAs
Public AAs
(1) (2)
(1) (2)
(1) (2)
(Friend)
{PHR, current illnesses, diagnosis}
{Hospital A, physician, M.D., internal medicine}
{Insurance company, Actuary, health insurance}

III. Types Of Tests

Unit Testing:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration Testing:

Integration tests are designed to test integrated software components to determine if they actually run as one program. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional Testing:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.
Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked. System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Strategy And Approach

Field testing will be performed manually and functional tests will be written in detail.

Test Objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

IV. Conclusions:

In this paper, we have proposed a novel framework of secure sharing of personal health records in cloud computing. Considering partially trustworthy cloud servers, we argue that to fully realize the patient-centric concept, patients shall have complete control of their own privacy through encrypting their PHR files to allow fine-grained access. The framework addresses the unique challenges brought by multiple PHR owners and users, in that we greatly reduce the complexity of key management while enhance the privacy guarantees compared with previous works. We utilize ABE to encrypt the PHR data, so that patients can allow access not only by personal users, but also various users from public domains with different professional roles, qualifications and affiliations. Furthermore, we enhance an existing MA-ABE scheme to handle efficient and on-demand user revocation, and prove its security. Through implementation and simulation, we show that our solution is both scalable and fficient.

Reference:

[1]. Author Reza Curtmola† Juan Garay‡ Seny Kamara§ Rafail Ostrovsky "Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions".

[2]. Author –Jin Goheujin@cs.stanford.edu "Privacy-preserving personal health record using multi-authority attribute based encryption with revocation ".

[3]. See discussions, stats, and author profiles for this publication January 2008"Efficient signature schemes supporting redaction, pseudonym mization, and data deidenti fication".

[4]. Frank Kargl, Institute of Media Informatics, Elaine Lawrence, Martin Fischer, Yen Yang Lim University of Technology Sydney Broadway 2007 NSW, "Security, Privacy and Legal Issues in Pervasive EHealth Monitoring Systems".

[5]. Upkar Varshney Published online: 12 July 2007# Springer Science + Business Media, LLC 2007 "Pervasive Healthcare and Wireless Health Monitoring".

[6]. R´emi Bazin_, Alexander Schaub_, Omar asany and Lionel Bruniey _Department of Computer cience, ´ Ecole Polytechnique 91120 Palaiseau, France yUniversity of Lyon, CNRS INSA-Lyon, LIRIS, UMR5205, F-69621, France "A Decentralized Anonymity-Preserving Reputation System with Constant-time Score Retrieval".