Cloud-Assisted Mobile-Access of Health Data With Privacy and Audit ability

**Abstract:**

Motivated by the privacy issues, curbing the adoption of electronic healthcare systems and the wild success of cloud service models, we propose to build privacy into mobile healthcare systems with the help of the private cloud. Our system offers salient features including efficient key management, privacy-preserving data storage, and retrieval, especially for retrieval at emergencies, and audit ability for misusing health data. Specifically, we propose to integrate key management from pseudorandom number generator for unlink ability, a secure indexing method for privacy preserving keyword search which hides both search and access patterns based on redundancy, and integrate the concept of attribute based encryption with threshold signing for providing role-based access control with audit ability to prevent potential misbehavior, in both normal and emergency cases.

**Architecture diagram:**

****

**Existing system:**

e-healthcare systems are increasingly popular, a large amount of personal data for medical purpose are involved, and people start to realize that they would completely lose control over their personal information once it enters the cyberspace. According to the government website, around 8 million patients’ health information was leaked in the past two years. There are good reasons for keeping medical data private and limiting the access. An employer may decide not to hire someone with certain diseases. An insurance company may refuse to provide life insurance knowing the disease history of a patient.

**Proposed System:**

Outsourcing the computation to the cloud saves TC3 from buying and maintaining servers, and allows TC3 to take advantage of Amazon’s expertise to process and analyze data faster and more efficiently.

The proposed cloud-assisted mobile health networking is inspired by the power, flexibility, convenience, and cost efficiency of the cloud-based data/computation outsourcing paradigm. We introduce the private cloud which can be considered as a service offered to mobile users. The proposed solutions are built on the service model shown in Fig. 1. A software as a service(SaaS) provider provides private cloud services by using the infrastructure of the public cloud providers (e.g., Amazon, Google). Mobile users outsource data processing tasks to the private cloud which stores the processed results on the public cloud. The cloud-assisted service model supports the implementation of practical privacy mechanisms since intensive computation and storage can be shifted to the cloud, leaving mobile users with lightweight tasks.

**Implementation modules:**

1. Medical Information Privacy Assurance(MIPA)
2. Searchable Symmetric Encryption
3. Identity-Based Encryption
4. Attribute-Based Encryption
5. Security Requirements

**Medical Information Privacy Assurance(MIPA):**

Some early works on privacy protection for e-health data concentrate on the framework design, including the demonstration of the significance of privacy for e-health systems, the authentication based on existing wireless infrastructure, the role-based approach for access restrictions, etc. In particular, identity-based encryption (IBE) has been used for enforcing simple role-based cryptographic access control. Among the earliest efforts on e-health privacy, Medical Information Privacy Assurance (MIPA) pointed out the importance and unique challenges of medical information privacy, and the devastating privacy breach facts that resulted from insufficient supporting technology. MIPA was one of the first few projects that sought to develop privacy technology and privacy-protecting infrastructures to facilitate the development of a health information system, in which individuals can actively protect their personal information. Privacy-preserving health data storage is studied by Sun *et al.* , where patients encrypt their own health data and store it on a third-party server.

**Searchable Symmetric Encryption:**

SSE allows data owners to store encrypted documents on remote server, which is modeled as honest-but-curious party, and simultaneously provides away to search over the encrypted documents.

*Key Gen*(*s*): This function is used by the users to generate keys to initialize the scheme. It takes the security parameter *s*and outputs a secret key *K*.

*Build Idx* (*D,K*): The user runs this function to build the indexes, denoted by *I*, for a collection of document *D*. It takes the secret key *K* and *D* and outputs *I*, through which document can be searchable while remaining encrypted.

*Trapdoor*(*K ,w*): The user runs this function to compute a trapdoor for a keyword *w*, enabling searching for this keyword. A trapdoor *Tw* can also be interpreted as a proxy for *w* in order to hide the real meaning of *w*. Therefore, *Tw* should leak the information about *w* as little as possible. The function takes the secret key *K* and the keyword *w* and outputs the respective trapdoor *Tw* .

*Search*(*I, Tw* ): This function is executed by the remote server to search for documents containing the user defined keyword *w*. Due to the use of the trapdoor, the server is able to carry out the specific query without knowing the real keyword. The function takes the built secure index *I* and the trapdoor *Tw* , and outputs the identifiers of files which contains keyword *w*.

**Identity-Based Encryption**

A practical IBE scheme in the random oracle model was proposed by Bone h and Franklin . Identity-based systems allow any party to generate a public key from a known identity value, for example, the string “alice@xyz.com” for Alice. IBE makes it possible for any party to encrypt message with no prior distribution of keys between individuals. It is an important application of the pairing-based cryptography.

**Attribute-Based Encryption:**

ABE has shown its promising future in fine-grained access control for outsourced sensitive data. Typically, data are encrypted by the owner under a set of attributes. The parties accessing the data are assigned access structures by the owner and can decrypt the data only if the access structures match the data attributes.

**Security Requirements:**

1) *Storage Privacy:* Storage on the public cloud is subject to five privacy requirements.

a) *Data confidentiality:* unauthorized parties (e.g., public cloud and outside attackers) should not learn the

content of the stored data.

b) *Anonymity:* no particular user can be associated with the storage and retrieval process, i.e., these processes

should be anonymous.

c) *Unlink ability:* unauthorized parties should not be able to link multiple data files to profile a user. It indicates

that the file identifiers should appear random and leak no useful information.

d) *Keyword privacy:* the keyword used for search should remain confidential because it may contain

sensitive information, which will prevent the public cloud from searching for the desired data files.

e) *Search pattern privacy:* whether the searches were for the same keyword or not, and the access pattern,

i.e., the set of documents that contain a keyword, should not be revealed. This requirement is the most challenging and none of the existing efficient SSE can satisfy it. It represents stronger privacy which is particularly needed for

highly sensitive applications like health data networks.

2) *Audit ability:* In emergency data access, the users may be physically unable to grant data access or without the perfect knowledge to decide if the data requester is a legitimate EMT. We require authorization to be fine-grained

and authorized parties’ access activities to leave a cryptographic evidence.

# System Configuration:-

# H/W System Configuration:-

# Processor - Pentium –III

**Speed - 1.1 Ghz**

**RAM - 256 MB(min)**

**Hard Disk - 20 GB**

**Floppy Drive - 1.44 MB**

**Key Board - Standard Windows Keyboard**

**Mouse - Two or Three Button Mouse**

**Monitor - SVGA**

# S/W System Configuration:-

* Operating System :Windows95/98/2000/XP
* Front End : java, jdk1.6
* Database : My SQL
* Database Connectivity : JDBC.