Bringing Healthcare to the Cloud

Security and Privacy Challenges

Benedikt Wolters
RWTH Aachen University

ABSTRACT

While in the past most electronic medical record (EMR) systems were used with only local usage in mind and electronic health records (EHR) were exchanged in a centralized one-to-one manner in medical applications, in recent years sharing of EHR in online healthcare portals has attracted a lot of attention. With Cloud Computing becoming more and more popular, healthcare providers and patients tend to shift their EHR data into the cloud. Cloud Computing offers scalability, resilience, adaptability, connectivity, cost reduction, and high performance features essential for a scalable critical EHR system. Despite the fact that sharing EHR data in the cloud can be beneficial for all included parties, it also poses several demands in preserving privacy and challenges in data security to prevent this very sensitive data from being leaked or exposed among third parties in the cloud with unknown, commercial or potential malicious interests.

In this paper we identify and discuss the privacy and security issues for shared EHR data in the cloud application context. We stress the specific peculiarities of EHR in contrast to other data. Subsequently, we formalize the requirements for patient-centric cloud EHR systems and we put forward cryptographic access control. Finally, we advert and compare two approaches which follow different encryption methodology.

1. INTRODUCTION

In February 2009, U.S. President Barack Obama signed the American Recovery and Reinvestment Act, which stated the American federal government to spend 19 billion dollar to digitize U.S. health records. His intention was to put forward the digitization of US health records, thereby creating jobs, reducing medical errors and bureaucratic overhead [6]. EHR systems promise to increase the efficiency and effectiveness of such electronic healthcare systems. The purpose of EHR services is to provide continuous and uninterrupted healthcare document access and to simplify the administration of healthcare records among different service domains such as different clinicians. Today many healthcare providers and health insurance companies already have implemented some form of electronic medial record (EMR) system. However, most of the records currently are stored in centralized database systems [18]. Traditionally, a patient has more than just one primary healthcare provider such as specialists, therapists, dentists, or other practitioners. Simultaneously, there can be various insurances for example additional dental or vision insurances. Additionally, other legitimate domains, e.g. family members or researchers have an interest in accessing those patient data [18]. With each healthcare provider typically having an own centralized database, EHR records are widely scattered among each provider. Hence there is a need for sharing data of EMR systems between providers and patients in different trust and administrative domains. Previous interoperability and sharing approaches between EMR systems by exchanging EHR have been reported to be a very tedious process which is also related to high costs and poor usability [21, 25, 16]. Furthermore, the insufficient interoperability reduces the value of an EHR [16] as it might lack important patient history. Especially in emergency situations immediate complete exchange of patients' EHRs is essential to save lives. Those limitations have been reported to restrict the distribution of health IT and especially EHR systems [21]. An open ubiquitous infrastructure platform could resolve this situation.

1.1 Appeal of Cloud Computing in Healthcare Applications

Cloud computing has become a promising paradigm gaining a lot of attention in recent years. The location of computing infrastructure is being shifted to third-party service providers that handle elastic management of hardware and software resources. Prominent examples for healthcare cloud providers are *Microsoft Health Vault* [3] or *Dossia* [1]. Moreover, large employers such as Intel, Wal-Mart, Applied Materials, and others, have committed millions of dollars to create a web-based framework that will supply over five million of their employees with access to personal health data through a common open-source architecture framework [16]. Cloud computing infrastructures deliver information technology as services, by facilitating the renting of IT resources such as software, database, storage, or computing power.

Therefore cloud computing provides an attractive enterprise IT platform to cut down the cost of EHR systems, while providing high scalability and availability and reducing the IT maintenance effort for medical staff. Cloud computing can not only increase the efficiency and costs of EHR sharing and management, it also enables ubiquitous access to healthcare services to the nomadic user anywhere at any time [12]. By offering a virtual infinite and elastic storage instead of building separate specialized data centers, in particular small care delivery organizations (CDO) can reduce high IT costs. Through the elasticity and usage-based pricing model it is possible to cut down the costs to a *pay-as-*

you-go manner, meaning paying only for the resources that are being utilized instead of the maximum capacity [12]. For example if a CDO requires 500 servers at peak in the noon but only can make use of 100 servers at night traditionally an organization would pay for $500 \cdot 24 = 12000$ server-hours instead of only for the average utilized $300 \cdot 24 = 7200$ server-hours not counting the costs for amortizing the IT infrastructure or additional operational costs [12]. Furthermore, using cloud infrastructure it is easy to add or remove needed resources leading to high adaptability.

1.2 Security and Privacy Issues

While it seems very attracting to store EHR data in the cloud, the patients' main concern are security and privacy risks: Studies have shown that ninety-one percent of people are strongly concerned about the privacy and security of their personal health information [16]. Obviously, e-health systems contain very sensitive private information and the leakage of any health related data can have severe consequences. As an example, banks or prospective employers could refuse a credit or a job position with insights into patient records. Furthermore, IT and healthcare providers would have to face intense legal penalties [19]. Being highly sensitive data, computerized medical records always have been open to abuse or threats. In file sharing networks thousands of accidentally published patient data records from many different sources can be found [15] containing not only sensitive medical data, but also financial information. Additionally, when storing patient data on cloud servers, patients and care providers loose physical control over their health data. There is the risk of sensitive patient data leakage by administrative personal of the cloud service providers. For example the U.S. Department of Veterans operates a Web-based PHR called MyHealtheVet, which allows US veterans to obtain authoritative health information, containing approximately 26.5 million military veteran information with social security numbers and health information [15]. Previously this database was stolen by an employee who had taken the data home without authorization [11, 18]. Moreover placing the data in a open ubiquitous cloud infrastructure, the systems are exposed to potential malicious outside attacks. If there is a successful security breach the data can be exposed. Beyond that, there is the need to grant access to subsets of patient records with a very fine granularity. For example a nurse might not have the same access rights to certain records as a therapist, or a patient wants to exclude information from a specific practitioner. On top of that there are government regulations such as the *U.S. Health Insurance Portability and Accountability Act* of 1996 (HIPAA) [4], which dictates minimal disclosure of patient data and confidentiality regulations. Unfortunately HIPAA only applies to "covered entities" such as CDOs. Organizations such as Microsoft (Microsoft Health Vault), Dossia or various other cloud service providers are not considered to be covered entities [16, 2]. In centralized approaches the primary method of guaranteeing privacy is implemented by enforcing classical data access policies in front of the data layer on the server side, in cloud computing with the cloud server not being trusted this is not possible [22]. A straightforward solution would be to encrypt the entire data record before uploading it to the cloud server. Conventional one-to-one encryption schemes however lack flexibility in sharing data and have potential high key management overhead which is not to be considered scalable [18]. So without appropriate security and privacy solutions especially designed for clouds, this potentially revolutionizing computing paradigm could become a huge failure. Several surveys of potential cloud adopters indicate that security and privacy is the primary concern preventing its wide adoption [22]. If this problems are solved, cloud PHR systems have the potential to protect patient privacy and security more than traditional paper-based records, it can provide additional security feature such as fine-grained access control, audit tracking or password protection [16].

1.3 Organization

The rest of this paper is organized as follows. First, we give an overview of different EHR/EMR systems and cloud schemes in Section 2. Secondly, we present an overview of the security and privacy issues in E-Health cloud infrastructure. Finally, in Section 4 we present two EHR cloud frameworks.

2. EHR/PHR SYSTEMS

Before discussing the privacy and security related issues we further specify types of electronic healthcare systems and cloud infrastructure.

2.1 Nomenclature

Allthough EHRs and EMRs are often used interchangeably by science, press, and healthcare industry [25] they describe different concepts in a narrow sense. We begin by further particularizing the terms and concepts of EMR, EHR, and PHR.

Electronic Medical Record is a legal record owned by a CDO. It includes what happened during a inpatient or outpatient visit or treatment of a patient. It contains no further information of other encounters at another CDOs. A EMR is being used, created and managed by healthcare staff within a CDO to document, supervise, and keep track of the healthcare delivery inside that CDO.

Electronic Health Record is a collection of EMR issued by different CDOs where the patient received services. In contrary to EMR it is created and owned by the patient and containts a longitudinal patient history as well as future care. EHR systems are typically run by community, state, or national emergence organizations [25].

Personal Health Record is a (electronic) health record that is created and managed by an individual. Ideally, a PHR containts a complete and full medical history of that individual. A PHR can consists of many sources such as EHR and EMR. PHR systems offer a wide variety of features such as the ability to exchange and view health data with other providers, scheduling appointments, renewing of prescriptions and more. The key aspect of a PHR is that it can be shared to others that have proper credentials to access it. By the definition of PHR it can be compared to a hub and spoke paradigm (Figure 1) [16, 23]. Data is aggregated from multiple sources (EMR/EHR, insurance, pharmacies, etc.). The patient-controlled PHR is in the middle and connected to all the sources and stakeholders providing information.

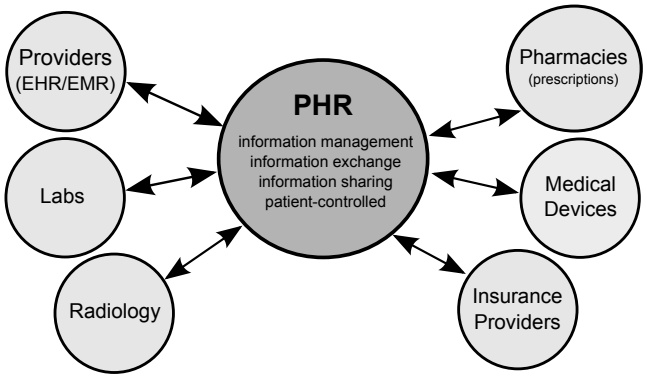


Figure 1: Hubs and Spoke Model for a PHR [16]. Multiple providers contribute to the patient-controlled PHR.

Due to privacy concerns many patients do not want to store their full PHR into EHR/EMR systems. However the overlap between data between EMR, EHR, and PHR systems differ from each patient according to her individual understanding of privacy [25]. A major distinction between EHR/EMR and PHR systems are legal requirements bound to the respective type of system: In most countries infrastructure that involves EHR/EMR is obligated to stricter security laws such as HIPAA or the German electronic Health Card (eHC) [19] while PHR systems generally are not.

Nevertheless, it is important to stress that each type of system introduced above can profit from utilizing the cloud computing paradigm. Additionally, aside from pure legal requirements PHR as well as EHR/EMR systems face fairly similar privacy and security challenges.

2.2 Cloud Taxonomy

We briefly introduce the levels for cloud infrastructures for a healthcare cloud system to illustrate the security responsibility tied to each level. We follow the classification given by Zhang et al. [25].

Software-as-a-Service (SaaS)

In this layer the cloud user is able to use the cloud service provider's applications which are running on cloud infrastructure. The customer does not control the underlying cloud services such as networking, server management, operating systems, but only uses client applications. Those applications can be accessible from different client platforms such as mobile devices or a web browser. On this layer the cloud service provider is in charge of security and privacy protection while the user typically is only in charge of security tokens such as passwords or single encryption keys.

Platform-as-a-Service (PaaS)

On the platform-as-a-service layer the consumer has the ability to deploy created applications written in programming languages supported by the cloud provider and utilizing precasted infrastructure or tools available. The consumer still has no control over server and underlying networking infrastructure, operating systems or detail insights into the storage system, but is fully in control of his application. In this layer two levels of security exist, one for the provider and the customer. The consumer is obligated to implement access control and authenticity requirements on the application layer, where on the cloud service provider may be responsible for lower level security mechanics such as end-to-end encryption [25].

Infrastructure-as-a-Service (IaaS)

In this layer cloud service providers offer the control over physical or virtual machines such as the choice of operating systems or machine configuration. The provider might also offer control of portions of networking components. On this layer a potential healthcare cloud customer is fully in charge of all operations concerning the security and privacy of his data records while the cloud provider simply provisions computing infrastructure.

3. PRIVACY AND SECURITY CHALLENGES IN CLOUD E-HEALTH SYSTEMS

In healthcare clouds many security and privacy challenges are correlated to the previous cloud taxonomy in use. We outline the challenges and security requirements for healthcare cloud systems.

3.1 Requirements and Challenges in Healthcare Clouds

If we think about the previous concepts of EHR/EMR and PHR the health records are stored distributively among several creators and owners. There are also multiple views of security. On the one hand there is the patient's view on the other the clinician's. Some of the medical records are stored in EHR/EMRs by CDOs after a visit, others such as historical records might be held by the patient herself or her relatives, family members etc. One major challenge for the patient is how to access and manage the control of the EMR records. Beyond that, a patient may only reveal specific or only designated records to family members or practitioners because of various patient-bound reasons. By the clinician's eye, there may exists patients in various EMR systems, depending on the type of sickness or specialty. One of the major challenges is how to securely access and store information. Another challenge is how to share patient records (with the patients-consent), for example, if a clinician wants to obtain a consolidation from a colleague. We now summarize the key privacy and security requirements for a healthcare cloud system.

"patient-centric" Privacy and Authorization

As we have already seen in the hubs and spoke model the patient is the key component of a health record. Ultimately, e-Health cloud systems such as PHR should give patients full control over the selective sharing of their own records. In patient-centric privacy [25], patients specify their own access policy per record. In this selective sharing policy patients should be allowed to specify access control at very fine granularity. In other words, various users can be authorized to access subsets of records due to patient-consent. Beyond that, there can be multiple writers such as healthcare professionals which can gain write-access to contribute information to PHR. It is not sufficient to simply grant access to a specific CDO in general as we expect that the whole CDO as a

organization, and thereby an unspecified set of entities, can not be trusted as such and single entities may misbehave. Therefore records should be addressable to individuals or personal groups with a specific role or certain attributes.

User Revocation

It should be possible to efficiently revoke a user's permission when necessary. For example after a finalized treatment episode or if an appointment by a certain health professional was not as well as anticipated, the patient should be able to remove the respective user privileges from her record. While this might sound trivial for upcoming future records once downloaded and saved to some doctor's local hard disk however it is hard to revoke access retrospectively. Nevertheless there exists approaches for DRM based EHR [14, 19] that can avoid this problem, but are not in scope of this paper. Apart from that data access policies should be flexible in a way that user defined rules should not only be revocable but also changeable and extensible.

Ownership of Information

The practice of storing locally or externally created data in the same system requires the identification of the owners and the origin of the information. It is necessary to store that information for the purpose of accuracy, responsibility in case of legal disputes [23]. There can be multiple roles regarding the ownership of a patient record [25]. The *owner* is generally known as the creator of the information, but can either refer to the *creator*, *author*, or *manager* of the information. The *creator* is the person or entity which is generating the data, typically this is a laboratory, radiology, medical imaging or a practitioner in healthcare. The *author* is the person which is responsible for the content. In medical systems the author is the CDO to which the creator belongs. A *manager* is the entity responsible for protection, sharing and distributing the information [23]. In patient-centric privacy the patient himself is referred to as the manager, while in decentralized (such as EMR) healthcare systems, manager may refer to a trusted third party, which is authorized by the patient or healthcare providers [18]. Nevertheless, it is not impossible in patient-centric privacy to have multiple owners such as relatives that also take over the manager role. Thereby a *multi-owner infrastructure* is required. Ownership of information can be achieved by utilizing a combination of encryption and watermarking techniques [25, 18].

Confidentiality

Confidentiality requires that information had only been made available to authorized users. This is made possible by cryptography. Authorized users possess enough credentials such as roles, attributes or the patients' clearance. Everybody else should not only be prevented from decrypting single records but also prevented to get an overview over a record's meta data or structural composition.

Authentication

Authentication describes the act of verifying one entity (i.e. a person or a technical system) claims of credentials and legitimation. Authentication in cloud environments is of high importance as data is becoming more accessible. Usually in centralized systems there is one authorization service per CDO or per system. Nonetheless in interoperable cloud healthcare services there is a need for issuing proper credentials to an entity such as the *American Medical Association* for healthcare staff, *American Board of Medical Specialties* for specializations, or the *American Hospital Association* for CDOs and insurance registries for issuing patient credentials [18, 23]. Those central registries can alter, grant, or revoke user accreditation. A common approach is to deploy smart cards with the corresponding cryptographic keys to each individual [10].

Authenticity, Non-repudiation and Data Integrity

Non-repudiation refers to the concept that any user in the system can obtain a secure proof which cannot be forged that confirms the identity of the owner and integrity of a data item. Furthermore, it must be ensured that neither party of a transaction cannot deny having received a data transaction or having performed a particular action related to the data [20]. This is usually achieved by using digital signatures for each involved party [25]. Along with non-repudiation comes the requirement of data integrity, not only ensuring the correct identity of a data issue but also guaranteeing that the accuracy and consistency of the record is tamper-proof. This is necessary for all included parties, for example a drug addict otherwise could tamper with prescriptions for a specific medicament or provide counterfeit certificates (such as attestation).

Availability

For the e-health cloud to function it must provide high availability. Service downtimes for example because of denial-of-service attacks, power outages or hardware failures are not acceptable as missing access to patient records might be life-threatening in an emergency scenario. Utilizing cloud computing paradigm most of this problems are being moved to cloud service providers. Proper failover solutions and backup strategies should be ensured and controlled in a periodically manner.

Notifications

When information is updated or new data is provided, all relevant parties including externals should be notified [23]. For example if a doctor submits discharge papers based on new medical test results into the healthcare systems all included parties should be notified according to the patient's consent that was defined. This requires the implementation of an interoperable notification framework including a registry of recipients. To prevent unwanted data disclosure it needs to be avoided that notifications are sent to third parties or entities, that are no longer responsible.

Audit Log

Audit is a tracking mechanism for controlling the integrity of the security infrastructure, which can be used to check for misbehavior or security breaches, but can also be used for review purposes to the patients or for fine-tuning existing access policies retrospectively [23]. Audit means recording all safety relevant user activities in an audit log in timestamped chronological order. Typically, user activities are access and modification of patient data. Additionally, in the audit log every new state of data should be put under version control to ensure that previous states can be reconstructed. This

is essential in a case of legal disputes to determine what information was accessible at what time [23].

Archiving
In some countries due to data retention laws and liability issues it is necessary to keep health information for a long timespan [23], which can overpower primary storage, but also can increase query complexity, since older information is not as relevant in day-to-day treatment. Archiving means moving healthcare information to secondary storage for reasons of system performance and primary storage capacity, but keeping it available in the case the data is needed. In the cloud context this means moving data to offline storage but being able to quickly restore it without any loss and to move it back to online storage when necessary [25].

3.2 Usability Tradeoffs

While fine-grained access is desirable, it also puts the burden on the patient to decide who should have the proper access rights and who should not. Non tech-savvy or elderly patients might be overwhelmed by the number of choices. To prevent imprudent users from being exposed the system should easily navigate the patients through their choices and should set up restrictive but practical default policies [6].

3.3 Emergency Access

In case of an emergency it might be necessary to circumvent the regular access policies due to urgency or unconsciousness. Therefore a break-glass access [18] is needed ignoring regular access policies. Since the patients' records are encrypted in a multi-owner scenario the patient could previously have delegated access control to a trusted emergency department (ED). Once an accident occurs the emergency staff involved has to contact the ED and the ED has to verify the staff personal and the event of an emergency. Then the emergency staff can obtain temporary keys to decrypt the patients records. After an emergency the patient can remove the temporary keys by revocation [18]. Another solution is to provide the patients with a "wear or carry medic-alert bracelet" [6]. This bracelet would contain the necessary decryption keys, but is sealed in a tamper-evident manner. Once the seal has been broken the key can be obtained from the bracelet, ensuring a emergence security breach can be recognized apart from the audit log.

3.4 Key Escrow Agents

In case a patient has lost her smart-card containing the proper encryption keys or forgot her key-passphrase it should be guaranteed that the data is not lost beyond repair. Another example could be a court verdict that dictates disclosure due to mental incapacity or imprisonment of a patient. Therefore similar to the emergency access, however depending on jurisdiction, cloud service providers may be obligated to provide decryption keys in such a scenario. While this is a controversial topic due to the potential of misuse [5] keys might be also stored by trusted third-party key escrow agents [6]. A key escrow agent is an additional entity that has the ability to decrypt a patients record, but is expected to be trustworthy. Nevertheless, to prevent abuse a threshold scheme [26] can be applied to enhance security against individual misbehavior. In a (t, n)-threshold secret sharing scheme there are n independent key escrow agents all holding chunks of the original key. This key, however, can only be reconstructed if t parties cooperate [26]. This ensures confidentiality and can prevent misbehavior of a single escrow agents.

In conclusion, we note that providing strong security and privacy in cloud infrastructures is a manifold topic. There are more challenges and requirements than just simple password protection or network security. In e-health cloud infrastructure security should be the major concern and guarding a cloud platforms is a multi-layer assignment which includes guaranteeing physical, network, application security, as well as secure data-backup strategy, and internal policies and procedures just as independent third-party certification [25].

4. ENCRYPTION AND ACCESS MANAGEMENT IN EHR SYSTEMS

In this section we discuss concepts on how to implement privacy in healthcare cloud applications. For this purpose we introduce two exemplary representative state-of-the-art schemes with cryptographic access control. The proposals by Benaloh et al. [6] and the work of Li et al. [18] have been subject to current research. Subsequently, we also compare the security of those schemes in terms of confidentiality, access control granularity, searchability, key management overhead and key revocation.

4.1 Traditional Access Control for EHRs

As we have pointed out before, traditionally full trust was placed on centralized servers where patients' data resides. In traditional EHR systems data is protected through access control models [18]. In the past different access control models have been proposed and applied to EHR systems. Prominent examples are Role-Based Access Control (RBAC) [23] and Attribute-Based Access Control (ABAC) [18]. In RBAC users access policies are granted according to their role. For example a clinician, a pharmacist, and a nurse could be roles linked with corresponding access rights. ABAC is extending the role concept to give users attributes. Each attribute holds specific partial access rights and access policies can be defined by declaring a set of attributes one must hold in order to access. In comparison ABAC is more favorable in the context of healthcare as it offers more flexibility in policy descriptions [18, 25], however for standardization and interoperability RBAC seems to be a de-facto standard [23].

While those models are not feasible for cloud computing [18], they are still noteworthy since the underlying concept of role or attribute based credentials is comparable to similar cryptographically enforces access control schemes for example Key Policy Attributed Based Encryption (KP-ABE) [13]. In the following we contemplate the cloud data server as not fully trustworthy but honest. That means the server provider will always try to find out as much as possible information but will honestly follow the proposed protocols. We further only focus on cryptographic access control techniques.

4.2 Patient Controlled Encryption

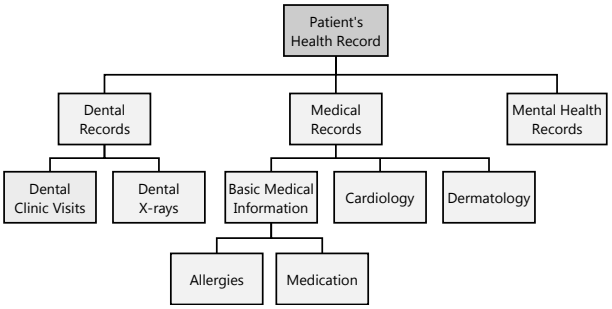


Figure 2: A sample patient's health record structure in PCE organized in a tree-based hierachy based on [6]. A patient can grant access to each subtree.

Benaloh et al. presented a hierarchical access control via a patient controlled encryption framework (PCE) [6]. EHRs are partitioned into a hierarchical structure. A patient is allowed to use her decryption key to derive subkeys for a specific hierarchical subset of the record. The patients are required to manage those subkeys. Subkeys can be given to other entities such as healthcare providers or family members, which can search and access the subsets of the record the subkey was designated for. The design principles of PCE are strong security and privacy while maintaining functionality through efficiency [6].

4.2.1 PCE Record Structure

It is assumed that a patients record is a collection of files, in which every file belongs to one category. Thus a patient's record is decomposed into mutual exclusive high-level categories. For example a patient record may contain the categories "Dental", "Mental Health" and "Medical Records" (Figure 2). Those categories can be further subdivided into specialized subcategories (e.g. "Dental Clinic Visits" and "Dental X-rays"). The tree-like hierarchical structure can be easily extended. The patient or parties which have access can create new subcategories within any existing category. Accordingly, within "Medication" (Figure 2) a practitioner could add a new category "Antibiotics", which other doctors with access to "Medication" can see directly the new section. Furthermore, each file in the patients record contains a filename, the name of the parent category it belongs to, a random "locator tag", a set of encrypted keywords, and the encrypted file itself. The filename and category name will be also encrypted to avoid disclosure. The locator tag will not reveal any information about the content of the file, but is used to identify files (see Section 4.2.3).

4.2.2 PCE Access Algorithms

The PCE framework as of [6] consists of four basic algorithms that handle cryptographic access, key generation and key derivation. The authors proposed diverging versions of the algorithms for public and symmetric key schemes (see Section 4.2.5).

Key generation algorithm that generates a secret root key and a public key for the patient. The root key is to decrypt the entire record or derivate keys for subcategories under the root category.

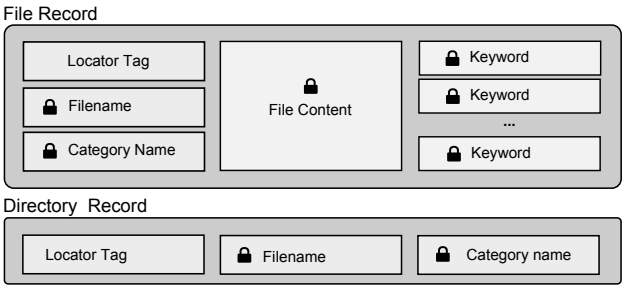


Figure 3: Records stored on the server in PCE. The curious (cloud) server will not be able to determine any information about the records.

Key derivation algorithm which takes a secret key for a category (or the root key in case of the root category), the name of the category as input and outputs a secret key for the specified category. This algorithm can be used by the patient to generate subkeys for doctors, but also by the doctor himself to generate subkeys for subcategories she has access to.

Encryption This algorithm takes a secret key (or in the case of a public key scheme a public key) for a category, and a file to be encrypted and outputs the ciphertext for the specified file under the specific category.

Decryption which takes the name of a category, the category's secret key and a ciphertext which was encrypted for that category and returns the decrypted file.

Using those algorithms a health data server will only retrieve encrypted files. A patient will run the key derivation algorithm for a specific category if she wishes a doctor to have access to that category. For the public key version Hierarchical Identity Based Encryption (HIBE) [8] scheme is used, which is based on Identity Based Encryption [6]. HIBE allows the encryptor to encrypt a message using hierarchical tuples of identities $(id_1, id_2, \ldots, id_k)$, which represent the user's position in the hierarchy. By his key a user can extract subkeys to any hierarchy level below his own $(id_1, id_2, \ldots, id_k, id_{k+1})$. In PCE this is adapted by declaring the category hierarchy as identities. For example a user holding a key for the hierarchy (*"Medical Records", "Basic Medical Information"*) will be able to generate subkeys for the "Allergies" and "Medication" category but not "Mental Health Records" (see Figure 2). For symmetric key version a pseudorandom generator function for polynomials is used to derive subkeys for each category. For details on this we refer to the original paper [6].

4.2.3 Directory Structure

In order to provide a directory of available contents in the patient record to any user at some point a category names need to be shown. If a doctor wants to derive a subkey for a category, he needs the decrypted category name. At the same time showing the labels of categories one has no access to discloses information. For example if there are many encrypted files stored in a directory called "Cancer", one can infer the patient has cancer [6].

Furthermore, the server and the users need to be able to refer to a file without revealing any content to the server. A user could download the entire record and then find out which portions he is able to encrypt. But this approach is not considered to be efficient. The goal is to provide a directory index that a user is partially able to decrypt according to his specific subkey, in order to gain access to the locator tags of a file, which do not reveal any information about the file, which then can be send to the server in order to download the file without the server learning anything about the file. When a party uploads an encrypted document to the server, a directory entry is also uploaded containing the locator tag, the category name and the file name. This directory entry will encrypted that only parties allowed to access that category can decrypt the directory entry (Figure 3). So a user can download the entire directory and try to decrypt each entry, if she successfully decrypts an entry she will find out the corresponding locator tag and send it to the server in order to request that document. The algorithm for encrypting the directory entries is similar to the one described in Section 4.2.2.

4.2.4 Searchability

The PCE framework proposes a searching mechanism in order to be able to query the server for keywords without the server being able to reveal what has been queried or the relationship between document and their keywords. When a doctor uploads health records to the server it is assumed that she also contribute a set of keywords, which are each also encrypted. It is proposed that the query is encrypted before sending it to the server. The server will be presented with an encrypted query and a set of encrypted keywords per file and will only be capable of determining whether a file matches an encrypted query or not. However a user can only search those categories which he is allowed to access. It is suggested that when performing the query a word cluster algorithm should be used to also encrypt similar related terms that could match to optimize search results. In the public key variant Searchable Public Key Encryption with Keyword Search (PEKS) is combined with HIBE [6]. The decryption key for a category is used to generate a trapdoor that allows the server to check whether a keyword is matching or not [6]. The symmetric key variant uses symmetric key searchable encryption (SSE). In SSE a encrypted keyword index is stored for each category so that the user possessing the appropriate key for that category generate trapdoors [6].

4.2.5 Public Key PCE vs. Symmetric Key PCE

Public key and symmetric key solutions have been introduced for PCE. In a public key implementation anyone who can obtain a patients public key and has upload rights can add documents to a patient's record without the overhead of retrieving special decryption keys. In spite of that, public key efficiency is in general much slower than symmetric key variants [6]. Also the public key option have a loss of privacy to some extend that a user (such as a server administrator) who has access to a patient public key and his history of search query trapdoors can use those trapdoors to determine what keyword has been searched for. This is done by trying out encrypting arbitrarily common keywords and testing whether the trapdoor matches a keyword or not [6]. In a symmetric key solution there is generally no key distinction between encryption and decryption, so anyone that can encrypt can simultaneously decrypt. This is not desirable in the case of an untrusted uploader for example a medical device that cannot guarantee the non-disclosure of the key. However, the symmetric key solution is more efficient [6] and solves the privacy issue in searchability by design as there are no public keys available and in order to perform a keyword search the appropriate secret keys have to be obtained.

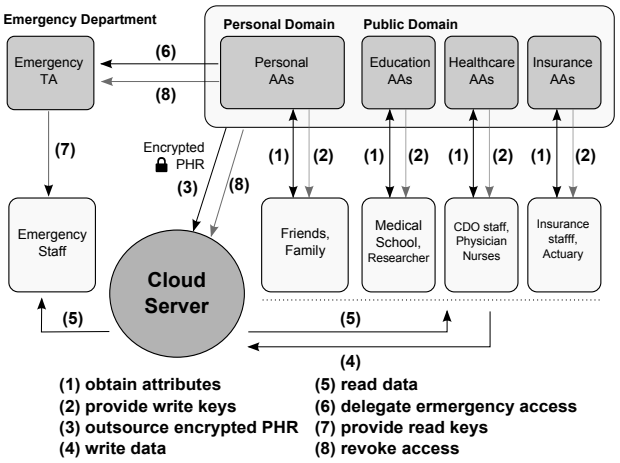


Figure 4: Framework architecture proposed by Li et al. [18]. Multiple user belonging to different domains can access and write to the cloud server.

4.3 Patient-Centric, Multi-Owner Data Access Control Framework

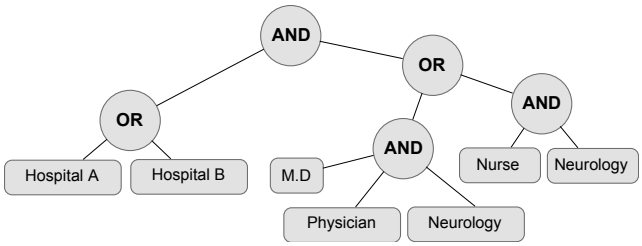
Li et al. [18] introduced a framework for achieving fine-grained access in large scale systems with many users. In such systems many users from different organizations, which can either be patients, their relatives, or healthcare personal from various CDOs, want to access different PHRs and therefore encrypt or decrypt according to their given permissions (e.g. in a manner that PCE suggested). Those users may come from several domains and may not even have a direct connection to the patient. This is called a *multi-owner* scenario. In a straightforward approach, this problem can simply be transformed to a key management issue, which requires a lot of key administration and user interaction with the owner of a record. However, assuming a large number of users, the patient will not only be overwhelmed by the linear increase of key management effort for each user [18], but also the accessibility and efficiency of such a system will suffer. The patient has to be always available to issue keys and manage new users, since encryption is done in a one-to-one manner. Typically, such a scenario is circumvented by deploying a central authority (CA) that could manage the keys, but in a healthcare cloud this would result in a loss of privacy, since the CA would have all encryption privileges. Therefore, Li et al. decompose the system into multiple security domains and apply variations of *Attribute Based Encryption* (ABE) [13] primitive to achieve fine-grained access control while simultaneously reducing the key managing overhead.

4.3.1 Architecture / Trust Authorities

The system is divided into multiple security domains. On the one hand there are users who have personal connections to the patient and can therefore be placed in a *Personal Domain* (PSD). On the other hand there are *Public Domains* (PUDs) such as insurances, hospitals, government or research facilities, where no personal connection exists. A PUD usually contains a large amount of users. Patients can encrypt their data so that both users from their PSD and PUD are able to read or write to it. Users from a PUD do not have to interact with the patient in order to gain access privileges but need to obtain credentials from their PUDs. Since users of a PUDs do not have to get in touch with the patient directly key management overhead for each user is highly reduced [18]. The architecture is depicted in Figure 4 showing multiple PUDs writing and accessing to a PHR.

4.3.2 Attribute Based Encryption

The central issue is how to allow the patients to specify their access policies for a file without direct need to know every user. Furthermore, even without knowing the user a strong privacy guarantee must be ensured. Attribute Based Encryption (ABE) [13, 18, 7] allows to encrypt a file in a one-to-many manner, where only users which have proper combination of attributes can decrypt it. ABE allows the patients to encrypt to a more general set of attributes a person can posses instead of a set of users that are known in advance. The attributes are combined in a tree access structure, where the leaves are attributes and the interior nodes consists of `AND` and `OR` gates. A user with set of attributes that can satisfy [13] the access-tree structure can decrypt the record. Figure 5 depicts an example access tree structure using attributes. Ciphertext-Policy ABE (CP-ABE) [7] allows that there is a central *Attribute Authority* (AA), which would define a public set of attributes to which an owner could encrypt the data using the public keys for the attributes of the AA. In CP-ABE, each user's private key is generated by a set of attributes she is possessing, and a ciphertext is encrypted with an access structure such that only users whose private key attributes satisfy a certain policy can decrypt it. However, we are dealing with a multi-authority scenario since e.g. there are a variety of hospitals in the healthcare PUD so there is no central authority, but many independent authorities. The use of classical CP-ABE would require that if patients want to encrypt a file for multiple AAs, they must upload multiple sets of ciphertext encrypted by the public keys and attributes of each AA. This is not only inefficient it also brings the risk of disclosure. For example if a file was encrypted for multiple hospitals and a single hospital misbehaves it can decrypt all of files for users of that hospital. Therefore two adapted variants of ABE are used. For every PUD (e.g. healthcare, research, education) a *Multi-Authority ABE* (MA-ABE) scheme [9] is adopted. MA-ABE is an extension to CP-ABE, which allows to have more than one central AA but to have multiple AA that coexists. The advantage to have multiple AAs is that Attributes can be assinged by independant AAs. That means AAs are now no longer entities such as hospitals, but organizations. For example an Organization may be the American Hospital Association, which could issue attributes for each hospital or the American Board of Medical Specialties, which could issue the specialty attributes for a practitioner. In MA-ABE each AA administer a disjoint set of attributes distributively.



$\mathcal{P}$:= (“Hospital A” ∨ “Hospital B”) ∧ ((“M.D.” ∧ “Physican”∧“Neorology”) ∨ (“Nurse”∧“Neurology”))

Figure 5: An MA-ABE access policy $\mathcal{P}$ specifying access control of a record for healthcare personal through AND/OR threshold gates [18]

.

Assuming there are N AAs within a PDU. Then the MA-ABE scheme ensures that any coalition up $N-2$ AAs can not break the security by colluding. For each PSD the standard KP-ABE [13, 7] scheme is adopted. In KP-ABE as also in MA-ABE, ciphertexts are associated with attributes, while user secret keys are defined with access structures on attributes [18].

4.3.3 Attribute Types and Key Distribution

In this architecture the owners (patients) distribute their access keys directly only to users in their PSD. The keys within a PUD are distributed by the respective AA. For example in Figure 4 an AA of an insurance company would issue a key to one of their actuaries. It is distinguished between data-attributes and role-attributes. For KP-ABE and the PSDs *data attributes* are specified and associated with the ciphertext. Data-attributes classify what types of files a user can access, also they can form an implicit hierarchical structure as in PCE. For example the set of attributes {“PHR”,“Medical History”,“Influenza History”} forms a hierarchical structure by attributes. The convention is that all attributes from the root (“PHR”) of the hierarchy to leaf node (“Influenza History”) should be included and combined by `OR` gates in order to allow each user with a specific attribute for a subset of that hierarchy access to a file. In our example a user with the attribute “Medical History” could access the record. Therefore ABE in terms of flexibility in access policy is equipotent to PCE. However, for the PUDs the owner does not encrypt to the users but to the attributes an AA holds. Hence *role attributes* and extra conventions are needed. Role attributes characterizes roles of an entity in an AA. There can be multiple role attribute types e.g. the corresponding department of a doctor, her medical licence, her CDO affiliation, or her medical specialty. For each of those attribute types the convention is that the owner needs to include at least one attribute. Additionally, there for each type of role attribute is a *wildcard attribute* “*” available, which symbolizes that the user does not care who can have access from that type. The wildcard attribute for each type is distributed to every user.

4.3.4 Emergency Access

It is assumed that the user has given his access rights to an emergency department (ED, Figure 4) beforehand. The emergency personal who wants to obtain access can then

Table 1: Framework Comparison

	PCE [6]	Li et al. [18]
Policy Granularity	by hierarchy inheritance of a record	hierarchal for PSD (KP-ABE) role-based for PUDs (MA-ABE)
Policy Flexibility	hierarchy is fixed but extendable to support multiple hierarchies	arbitrary attribute access policies can be combined through `OR` gates
User Revocation	complete re-encrytpion necessary	lazy-revocation, successive proxy re-encryption
Key Escrow Agent	no support (tamper-proof bracelet containing secret key)	ED Delegation
Searchability	included in the subliminal proposal, limited expressiveness	can be extended by APKS[17] also equality, subset, and range queries
Ownership	only the patient	multi-owner support (PUDs/PSDs)
Key Management	linear complexity per user	adaptable through ABE

get in contact with the ED and authenticate. Also, the ED needs to verify the emergency scenario in order to prevent abuse. Subsequently the ED can issue temporary write keys, which the patient can revoke after the emergency situation.

4.3.5 User Revocation

When a user's attribute change or a patient denies a user access to his record a user needs to be revoked. In case of the PUD the AAs revoke the user's role attributes, which results in the loss of read access. Traditional revocation schemes would require users that were not revoked to obtain key updates [18]. As this is not efficient only the public key as well as the secret key components for the affected attributes should be updated. Furthermore, the ciphertexts components affected by those attributes need to be updated. This operation can be partially shifted to the cloud server by utilizing lazy-revocation and proxy re-encryption. *Proxy re-encryption* allows that a cloud server can be given a re-encryption key $r_{a \leftrightarrow b}$. The server can then translate ciphertexts under the public key p_a into new ciphertexts under the public key p_b [24]. Additionally, a version number i is attached and deployed with each attribute. So when an revocation event occurs, the owner (or the AA) submits a re-encryption key and increases the version number of that attribute [18]. *Lazy-revocation* describes to only update the affected ciphertexts and user secret keys, when a user logs into the system. A user can compare his attribute version number against the current one and retrieve the aggregated key updates that have occurred since his last login [18].

4.3.6 Granting (Temporary) Write Access

If there is no control over the write access everybody could encrypt to attributes of an owner's record. In a straightforward solution an organization such as an AA would issue write signatures, every time a user wants to perform a write action. Yet, this requires the organization to be always to online. Li et al. propose that the organization defines a *working cycle* (e.g. one day). For every working cycle and a time granularity Δt a hash chain $\mathcal{H} := (h_0, h_1, \ldots, h_n), H(h_{i-1}) = h_i$ is generated where H is a cryptographic hash function. A signature with the end of the hash chain h_n is broadcasted by the organization. After each time period i the organization multicasts h_{n-i} to the authorized users, but not the revoked users. Furthermore, a record patient needs to deploy a time-related signature (ts, tt) with start (ts) and end (tt) timestamps. When a user tries to upload a document the cloud server at time j he includes $(ts, tt), (h_{n-j})$ and h_n can then check the signatures, the owner's time period, and the integrity of the hash chain by evaluating $H^j(h_{n-j}) = h_n$,where H^j is the j-fold composition of H. If everything holds, write access is granted [18].

4.4 Framework Comparison

We summarize our findings in Table 1. In PCE access policies are defined through the subkey of a category which is attached to a fixed hierarchy. Although Benaloh et al. [6] propose an extension where an unlimited number of hierarchies can coexist with no significant loss in efficiency, the fact that one has to abide to the hierarchical still lacks a fair amount of flexibility. The framework proposed by Li et al. offers more flexibility in specifying the access policies because attributes can be assigned arbitrarily. Nevertheless, it has to be noted that in PCE it is decently easy to create new hierarchal subcategories while in Li et al. attributes that PUDs govern are usually predefined and fixed. Thereby the the expressibility of the encryptor's access policy is also limited, as he is restricted to the set of attributes that the PUDs offer. However, the key difference between both frameworks is the potential key management overhead. In a large PHRs there are typically a lot of users. Thereby PCE suffers from a high key management overhead, since for each user a key has to be generated. With the use of an ABE scheme, the access policies are defined through combination and attributes, which reduces the key management complexity already by design. The support of multiple authorities in Li et al. further reduces key management complexity, since distribution of attribute keys is delegated to the PUDs.
When it comes to user-revocation PCE requires a complete re-encryption, while Li et. al utilize proxy re-encryption, which can can be partially delegated to the cloud server. Furthermore, Li et al. use lazy-revocation to aggregate multiple ciphertext updates on demand. Another upcoming requirement for e-health cloud frameworks is searchability over encrypted records. Even though the original PCE proposal includes searchability without leaking privacy the expressiveness is limited to equality queries. The framework of Li et al. can be extended by Authorized Private Keyword Searches (APKS) [17] which also supports equality, subset and range queries under fine-grained access control and preservation of privacy.

5. CONCLUSION

First we adverted the concept of cloud computing to be a practical and beneficial approach to storing patient data. We identified the challenges for building and conducting in security and privacy for electronic healthcare systems in cloud-based environments. In the meantime, we have examined important concepts that relate to patient information sharing and dissected security and privacy issues that arise especially in access and administration of EMR/EHRs and PHRs. Subsequently, we introduced two varying approaches that implement security and privacy in cloud environments through cryptographic enforced access policies. This approaches which are uniquely designed for addressing the previous challenges demonstrate that the cloud computing paradigm can be utilized for securely storing medical records. Future implementations will show whether the presented frameworks are feasible in a practical scenario and whether user will subscribe to the concept of securely storing patient data by utilizing cloud computing.

6. REFERENCES

[1] Dossia Personal Health Plattform. `http://www.dossia.org/` [accessed 30-March-2013].

[2] Google, Microsoft Say HIPAA Stimulus Rule Doesn't Apply to Them. `http://www.ihealthbeat.org/Articles/2009/4/8/Google-Microsoft-Say-HIPAA-Stimulus-Provision-Doesnt-Apply-to-Them.aspx` [accessed 03-April-2013].

[3] Microsoft Health Vault. `http://www.healthvault.com` [accessed 30-March-2013].

[4] The Health Insurance Protability and Accountability Act of 1996 - Centers for Medicare and Medicaid Services. `http://www.cms.gov/Regulations-and-Guidance/HIPAA-Administrative-Simplification/HIPAAGenInfo/` [accessed 01-April-2013].

[5] H. Abelson, R. N. Anderson, S. M. Bellovin, J. Benaloh, M. Blaze, et al. The risks of key recovery, key escrow, and trusted third-party encryption. 1997.

[6] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter. Patient controlled encryption: ensuring privacy of electronic medical records. In *Proc. 2009 ACM workshop on Cloud Computing Sec.*, pages 103–114. ACM, 2009.

[7] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *Security and Privacy, IEEE Sym. on*, pages 321–334. IEEE, 2007.

[8] D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *Advances in Cryptology–EUROCRYPT 2005*, pages 440–456. Springer, 2005.

[9] M. Chase and S. S. Chow. Improving privacy and security in multi-authority attribute-based encryption. In *Proc. 16th ACM Conf. on Computer and Comm. Sec.*, pages 121–130. ACM, 2009.

[10] Y.-Y. Chen, J.-C. Lu, and J.-K. Jan. A secure EHR system based on hybrid clouds. *Jour. of Medical Systems*, 36(5):3375–3384, 2012.

[11] J. Foreman. At risk of exposure. *LA Times*, 2006. `http://articles.latimes.com/2006/jun/26/health/he-privacy26` [accessed 30-March-2013].

[12] A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica. Above the clouds: A berkeley view of cloud computing. *Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS*, 28, 2009.

[13] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proc. 13th ACM Conf Computer and Comm. Security*, pages 89–98. ACM, 2006.

[14] M. Jafari, R. Safavi-Naini, and N. P. Sheppard. A rights management approach to protection of privacy in a cloud of electronic health records. In *Proc. 11th Ann. ACM workshop on DRM*, pages 23–30. ACM, 2011.

[15] M. E. Johnson. Data hemorrhages in the health-care sector. In *Financial Cryptography and Data Security*, pages 71–89. Springer, 2009.

[16] D. C. Kaelber, A. K. Jha, D. Johnston, B. Middleton, and D. W. Bates. A research agenda for personal health records (phrs). *Jour. of the AMIA*, 15(6):729–736, 2008.

[17] M. Li, S. Yu, N. Cao, and W. Lou. Authorized private keyword search over encrypted data in cloud computing. In *Distributed Computing Systems (ICDCS), 2011 31st Int. Conf. on*, pages 383–392. IEEE, 2011.

[18] M. Li, S. Yu, K. Ren, and W. Lou. Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings. In *Security and Privacy in Comm. Networks*, pages 89–106. Springer, 2010.

[19] H. Löhr, A.-R. Sadeghi, and M. Winandy. Securing the e-health cloud. In *Proc. 1st ACM Int. Health Inf. Sym.*, pages 220–229. ACM, 2010.

[20] A. McCullagh and W. Caelli. Non-repudiation in the Digital Environment. *First Monday*, 5(8-7), 2000.

[21] P. Moore. Navigating the Tech Maze. *Physicians Practice*, 2009. `http://www.physicianspractice.com/articles/navigating-tech-maze` [accessed 30-March-2013].

[22] H. Takabi, J. B. Joshi, and G.-J. Ahn. Security and privacy challenges in cloud computing environments. *Security & Privacy, IEEE*, 8(6):24–31, 2010.

[23] H. van der Linden, D. Kalra, A. Hasman, and J. Talmon. Inter-organizational future proof ehr systems: a review of the security and privacy related issues. *Int. Jour. of Medical Inf.*, 78(3):141–160, 2009.

[24] S. Yu, C. Wang, K. Ren, and W. Lou. Attribute based data sharing with attribute revocation. In *Proc. 5th ACM Symp. on Information, Computer and Comm. Sec.*, pages 261–270. ACM, 2010.

[25] R. Zhang and L. Liu. Security models and requirements for healthcare application clouds. In *2010 IEEE 3rd Int. Conf. Cloud Computing*, pages 268–275. IEEE, 2010.

[26] W. Zhang and J. Gao. A threshold key escrow scheme for placing of escrow agent flexibly and identifying cheaters efficiently. In *Advanced Management Science (ICAMS), 2010 IEEE Int. Conf. on*, volume 3, pages 409–412, 2010.