Health Records and the Cloud Computing Paradigm from a Privacy Perspective

Christian Stingl*; Daniel Slamanig
Department of Medical Information Technology, Carinthia University of Applied Sciences (CUAS), Klagenfurt, Austria

Submitted July 2010. Accepted for publication June 2011.

ABSTRACT
With the advent of cloud computing, the realization of highly available electronic health records providing location-independent access seems to be very promising. However, cloud computing raises major security issues that need to be addressed particularly within the health care domain. The protection of the privacy of individuals often seems to be left on the sidelines. For instance, common protection against malicious insiders, i.e., non-disclosure agreements, is purely organizational. Clearly, such measures cannot prevent misuses but can at least discourage it. In this paper, we present an approach to storing highly sensitive health data in the cloud whereas the protection of patient's privacy is exclusively based on technical measures, so that users and providers of health records do not need to trust the cloud provider with privacy related issues. Our technical measures comprise anonymous communication and authentication, anonymous yet authorized transactions and pseudonymization of databases.

Keywords: electronic health record, personal health record, cloud computing, privacy aspects, privacy protection

1. INTRODUCTION

Cloud computing is an emerging computing paradigm which subsumes many existing concepts such as distributed, grid and utility computing, and focuses on providing infrastructure, platforms as well as software over the Internet. It is believed that the cloud will be the next generation infrastructure for hosting data, deploying software and services, and is already used in various fields of applications. In the field of electronic healthcare and especially in context of personal health records, major software vendors are currently adopting cloud technology. For instance, Google Health [1] and Microsoft's HealthVault [2] have already been launched, though still in beta phase.

*Corresponding Author: Christian Stingl, Department of Medical Information Technology, Carinthia University of Applied Sciences, Primoschgasse 10, 9020 Klagenfurt, Austria. Phone: +43 (0)5 90500-3233. Fax: +43 (0)5 90500-3210. E-mail: c.stingl@cuas.at. Other author: d.slamanig@cuas.at.

From a medical and economic point of view, these systems are very interesting, since they support location-independent access to medical data in treatment processes and avoid unnecessary multiple examinations. However, the protection of patient's privacy is, from our point of view, considered as of subordinate importance in current concepts, and providers are not aware of the associated risks. For instance, insiders (employees at the cloud provider) are usually considered as absolutely trustworthy, which is in contrast to previous studies [3]. In context of highly sensitive medical data, it is in our opinion inevitable to consider privacy protection as an integral part of the entire system. Especially in cloud computing, the warranty for high availability and scalability implies a large number of insiders as well as data replicas. Consequently, the attack surface can be assumed to increase when realizing health records in the cloud.

In this paper, we propose a concept for health records in the cloud, where the protection of patient's privacy is a fundamental design criterion. Thereby, the main idea is that the provider establishes basic security mechanism, e.g., firewalls, intrusion detection/prevention systems, physical access control, and hosts the health record system. This health record system is designed such that users can integrate the data according to a well defined conceptual model, but without disclosing any information concerning patient's health records. This means that a person, e.g., an insider or an intruder, who even gains full access to the entire health record system is not able to determine any information about the content and the structure of patient's health records. However, persons who are explicitly granted can efficiently access these records.

2. CLOUD COMPUTING

Currently, cloud computing which aims at transparently providing highly reliable and dynamically scalable resources and services on the Internet has become increasingly attractive. It is actually a buzzword incorporating many existing concepts such as distributed, grid and utility computing, but is extremely pushed by major corporate players like Amazon, Google and Microsoft in recent times. It is believed to be the next generation infrastructure for hosting data and deploying software and services; it is very likely that it will see wide adoption soon [4]. The basic idea behind cloud computing is that operators of large computing farms rent IT-related capabilities "as a service" to users or companies, by allowing them to access technology-enabled services, without any need for knowledge of, expertise with, or control over how the technology infrastructure that supports those services work. We refer the reader also to [5] for a detailed discussion of the basic terminology and concepts presented below and note that these terms have recently also been defined by the National Institute of Standards and Technology (NIST) in [6].

- Infrastructure as a Service (IaaS): IaaS refers to a provision model in which a cloud provider delivers infrastructure components such as CPU, memory and storage, typically realized via a platform virtualization environment (for running client-specified virtual machines), as a service. Thus, clients can obtain and boot new virtual server instances on demand, which allows to quickly scale capacity, both up and down, as the computing requirements change. Amazon's Elastic Compute Cloud (EC2) is a prominent example for an IaaS offer.

- Platform as a Service (PaaS): PaaS refers to a provision model in which a cloud provider offers a platform for building and running web-based applications. The PaaS model provides all of the facilities required to support the complete life cycle of building and delivering web applications and services entirely available from the Internet, all with no software downloads or installation for developers and end-users. An example for a PaaS is the Google App Engine, which is a platform for developing and hosting web applications in Google-managed data centers or Microsoft's Azure.
- Software as a Service (SaaS): SaaS refers to a provision model in which a cloud provider offers his/her clients ready-to-use applications through a subscription or a "pay as you go" model. A prominent example for Saas is Google Apps, which provides several web applications with similar functionality to traditional office suites (email, calendar, instant messaging, word processing, spreadsheets, etc.).

To access these cloud services, two main technologies can be currently identified. Web Services are commonly used to provide access to IaaS services and web browsers are used to access SaaS applications. In PaaS environments, both approaches can be found. Besides the above categorization regarding the type of service, it is also necessary to consider the place where these services are hosted (cf. [5]):

- Public cloud: The resources are provided by off-site third-party providers, e.g. Amazon.
- Private cloud: The resources are provided by the organization which uses the services itself.
- Hybrid cloud: Represents a combination of both approaches where for example sensitive data are stored in the private cloud and non-sensitive information is stored in the public cloud.

From a holistic point of view, cloud computing provides many advantages for all parties involved. Cloud providers can achieve a high capacity utilization of their computing farms leading to more cost efficiency. Service providers being customers of cloud providers no longer need to build and manage their own costly IT-infrastructures, but can use state of the art infrastructures at cloud providers on a "pay as you go" basis; i.e., they only pay for the resources (consumed storage space, data transfer, computing resources) they actually use. Finally, users can profit from moving and remotely storing any locally stored information, such as email, word processing documents, etc., into the "cloud". This provides enormous benefits to users, since they have ubiquitous access to their information from their computer, laptop or mobile device from any place in the world.

3. ELECTRONIC AND PERSONAL HEALTH RECORDS

The main idea of health records is to compile a lifelong documentation of all pertinent medical data of a person. This lifelong character of these records is essential, since many treatment processes require a more or less complete overview of the medical history of a person.

Health records are in general classified as electronic medical records (EMRs), electronic health records (EHRs) and personal health records (PHRs). Thereby, an EMR

includes medical records of patients, which are for instance managed by clinicians. An EHR integrates all pertinent health information of individuals from different medical institutions and these data are shared between them. PHRs are similar to EHRs, but intend that the patients manage or decide who should manage their health record [7]. Thereby, the management also includes integrating data in the health record. It should be noted that the system providing an EHR or PHR (denoted HR henceforth) is called EHR system and PHR system, respectively.

A relevant aspect for our further discussion is that especially in case of PHRs, individuals are able to access and manage their health information via the Internet. Additionally, individuals are able to share these data with other parties and can access data of other parties for whom they are authorized. We note that this useful functionality plays an important role when considering privacy protection in such systems.

Especially, the location-independent access to patient related data offers the most promising potential to improve the quality and efficiency of medical treatment processes. Besides this primary use of medical data, one additional feature of HRs is the so-called secondary use of data. It is related to non-direct care use of personal health information, including but not limited to analysis, research, quality and safety measurement, public health, payment, provider certification or accreditation, marketing and other business including strictly commercial activities. Nevertheless, in our opinion, HRs should mainly focus on the primary use of medical data to achieve the best possible benefits for the patients.

In contrast to the aforementioned advantages, HRs are not free of dangers. This is due to the fact that the central availability of health-related data results in an increased potential for the misuse of these data. This problem is often addressed by data protection officers as well as representatives of medical institutions, and major concerns are the so-called transparent patient and physician. The former means that the state of health of a person is fully transparent to parties who are able to obtain access to the HRs. The latter one denotes the transparency of all actions conducted by a specific physician.

4. CONTENT DATA AND METADATA IN CONTEXT OF HEALTH RECORDS

In this section, we will classify data contained in HRs. But firstly we note that the concrete realization of the HR is not that important, since we are only interested in the categories of data available in such systems. The most important distinction is between patient-related and non-patient-related data.

Non-patient-related data are usually the master data of the system, e.g., all involved clinics, employees, code-systems. We emphasize that these data considered on their own have absolutely no relation to patients.

Patient-related data can be classified as administrative data of the patient and medical data concerning the patient. Administrative data are a preliminary to establish a health record for a specific patient and to uniquely identify the patient within the system. Obviously, some of this information needs to be available to any party within the system. This means for instance that authorized users, such as physicians, can search for a particular patient but does not necessarily access the patient's health record. The second class of patient-related data consists on the one hand of structured medical

data (e.g., diagnoses, medication) which are usually stored in a database. These data define at least the structure of the health record of every single patient, in general contain all relevant medical details (e.g., diagnoses or measurement data), and are directly linked to the master data. On the other hand, especially in health records, there are many types of data which are not fully structured (usually represented by documents). Such documents are integrated in storage systems and linked to the health record.

Our main motivation for the distinction between structured and semi structured/unstructured medical data is that those two classes require different techniques to prevent misuse of these data. In case of structured data, the data itself does only contain a link to the patients, but in general no explicit patient identifying information. Hence, it is sufficient to obfuscate these links, which will be discussed in section 7. However, in the second class, documents contain explicit identifying information, e.g., the patient's name or social security number within the running text of a medical report. To protect these documents, the obfuscation of the relation between documents and the respective patients is necessary but not sufficient. In addition, the content of documents needs to be protected too. The details on technical issues to realize this will be discussed in section 7. A succinct presentation of the classification is the following:

- Metadata of the health record system: Basic metadata of a health record system, e.g., clinics, departments, physicians, etc.
- Administrative patient data: Administrative data regarding patients, e.g., name, address, gender, age, etc.
- Structured medical data: These data are linked to a patient in the health record, e.g., diagnoses, measurement data, etc. This is usually realized by means of relations in a relational database model.
- Semi structured or unstructured medical data: These medical data (e.g., medical findings) contain identifying information (e.g., XML documents, pdf documents, medical images) of the patients but are also linked to patients.

In general, it is only possible to identify a patient using structured medical data when having access to the relations and consequently to the administrative data of the patients. Moreover, semi structured and unstructured data also contain person-identifying attributes, e.g., patient's name and address.

Besides these medical data, when considering privacy protection in HR systems, it is also important to consider data that are not part of a patient's health record, e.g.,

- Medical staff accessing health records of patients: If an adversary is able to figure out that an oncologist accesses the health record of an individual, then this may reveal information about the state of health of the individual.
- Frequency of logins of a user: One may assume that the frequency of accesses or integration of data (into the HR) of users reflects their state of health in some way.
- Correlation between authenticated users and actions in the system: If a patient is authenticated (and thus has a session with the system), then every action within the system can be linked to the patient. For instance, if a patient inserts

medical data without identifying information, the data can be linked to the patient through the authentication.

- Data logged by other infrastructural components: For instance, network traffic logged by intrusion detection systems (IDS).

It is common sense that medical data are classified as sensitive. Moreover, a part of these data can be considered as highly sensitive because the unwarranted knowledge of this information can massively violate the patient's privacy, and may furthermore negatively influence the well-being of the patient. This is for instance reflected by a study [8] where human resource managers were asked to rank equally qualified job candidates suffering from different diseases. The results show that the more sensitive medical data, the less are the chances to get a job.

5. HEALTH RECORDS IN THE CLOUD

After having considered the concept of cloud computing and health records in a separate way, we will now discuss general issues that emerge or result from combining both technologies. As already mentioned above, when realizing a health record, it is inevitable that a storage service (for files) and a database service (for structured data) are provided. Thereby, it must be noted that the latter needs to provide all common features of a relational database management system. Simplified databases, which do not support complex schemes as well as transactions, are not sufficient from our point of view. Moreover, a health record requires a platform to implement a rich set of business logic functionalities, e.g., the management of users or the realization of access rights. Considering the concrete implementation of the aforementioned components in the cloud computing model, we identify the following two approaches from the perspective of a health record provider:

- Platform as a Service (PaaS): A health record provider implements his/her own health record using development tools and platform specific functionalities (database, identity management, access control, etc.) of the cloud provider. The costs to implement a health record clearly depend on the specific platform of the cloud provider, e.g., Microsoft's Azure.
- Software as a Service (SaaS): Considering this approach, a health record provider does not implement a health record on his/her own, but uses a health record service provided by a specific cloud provider, e.g., Google in case of Google Health.

However, from the point of view of an end user (e.g., patients or medical staff) both approaches are SaaS solutions. Hence, for our further discussion, it does not matter which of the above approaches is taken. Regarding the deployment model in cloud computing, only public clouds offer the relevant advantages of a health record, such as location-independent access to medical data. The use of private clouds does not seem to be suitable in context of health records which are cross enterprise solutions.

As mentioned in the INTRODUCTION, neither the cloud computing model nor health records are without controversy. The most important issues are listed below, although they are not specific to health records in the cloud. Nevertheless, due to the sensitive character of health data, their relevance increases.

- Availability: A health record must be available 24/7/365, since unavailability of the service could imply serious effects on the patient's health.
- Loss of data: In case of loss of sensitive data, it must be specified who is responsible for this loss and which subsequent actions have to be taken.
- User's acceptance: There are several psychological factors, such as uncertainty about the location of data, number of replicas, availability of data, etc., which may reduce the trust in and usage of a health record service.
- Legal issues: If the cloud provider, the health record provider and potentially the end users are from different countries, then legal issues have to be considered very carefully. It is getting even more complex if cloud providers are storing their replicas in different countries.
- Financial issues: If a cloud provider goes out of business, it is extremely uncertain how and if the health record service provider will get all his/her data back. Furthermore, it might also be impractical to restore large amounts of data from the cloud provider's site. The last aspect also applies when changing a cloud provider.
- Security issues: The cloud provider needs to establish suitable security measures to prevent attacks against or intrusion into the system by using, e.g., firewalls, intrusion detection and prevention systems.
- Privacy issues: Since medical data are very sensitive and their misuse could have serious consequences, the protection of these patient-related data is extremely important.

Considering all aforementioned issues, from an engineering perspective besides availability and data loss, the security and privacy issues are the most challenging and often mentioned in context of health records. Especially, privacy is referred to as the most critical factor of success by end users in surveys concerning health records. Availability and data loss are fundamental issues in context of cloud computing, which are more or less independent of the application scenario. Nevertheless, due to the lack of adoption of large scale applications, it still has to be shown in the future that these issues can be handled appropriately in practice by cloud providers.

In contrast, regarding security and especially privacy, neither precise goals that must be achieved nor the methods to achieve these goals are clear at all. In the remaining sections, we will focus on privacy aspects and propose methods to preserve privacy in health record systems.

6. CLOUD PRIVACY THREATS

In order to develop appropriate measures to protect patient's privacy, it is necessary to perform a detailed threat analysis including the integration of data into the health record system, and the transmission and storage of data as well as the access to these data with respect to potential adversaries and attack scenarios. In abstracting from the specific details of the individual processes, four main components are identified that could potentially be attacked to violate patient's privacy:

- Data input and representation: This component represents the client who provides functionalities to input, search, retrieve and visualize data. In the cloud computing model, this will typically be a web browser.

- Data transmission: This component represents the bidirectional transmission of data from the user's client to the health record system and vice versa. In context of public clouds, the transmission will be realized via the Internet.
- Data processing and distribution: This component receives client's data, sends data back to clients and represents the health record system. Moreover, this component is responsible for storing data in different locations. Using cloud computing, this will typically be a web application.
- Data storage: This component is responsible for the persistent storage of structured and unstructured data. In context of cloud computing, this can be realized by means of relational databases and cloud storage services, respectively.

There are two different classes of adversaries:

- Internal adversaries: These adversaries, such as employees of the cloud provider, conduct attacks mainly against the health record system. It is reflected in current studies that more than 50% of attacks against information systems are launched by insiders [7]. But surprisingly enough, insiders are often not considered as potential adversaries in designing security measures of systems, or in preventing insider attacks by means of organizational measures such as security policies and/or non-disclosure agreements. The aforementioned measures are necessary but not sufficient, since they can only discourage misuse but not prevent it at all.
- External adversaries: These adversaries are usually considered with respect to security for health record systems. These adversaries can either behave active, i.e. hackers, or passive, i.e. eavesdroppers. In contrast to passive adversaries who are only eavesdropping the communication channel, active adversaries also manipulate transferred and/or stored data.

Insiders can always obtain unauthorized access to medical data more easily than external adversaries. Therefore, a reliable security concept for a health record system needs to address insider attacks at least as carefully as external attacks. While both of these two types of adversaries can conduct attacks discussed below, insiders can usually mount attacks on the cloud system much easier than external adversaries.

- Attacks on the client component: The main focus of these attacks is to steal user's authentication credentials (e.g., username/password) or medical data accessed by a user via the user's client. Methods to launch these attacks can, for instance, be phishing, pharming and using several types of malware, e.g., Trojan horses, key loggers, etc., running on the user's client.
- Attacks on the transmission channel: The goals of attacks on the transmission channel are the same as above, but the risk is potentially higher since data of many different users may be stolen. The methods of these attacks are, however, different and may include, for instance, sniffing and person-in-the-middle attacks.
- Attacks on the cloud system: The main focus of these attacks is to reduce the availability of the system, and to manipulate or steal data. The first type of attack can be accomplished, for instance, by flooding attacks [9] to achieve denial of service (DoS). The latter attacks typically use exploits for

components of the cloud system, or are launched by insiders who abuse their privileges by, for example, copying files from the storage or stealing a copy of a database.

With respect to data theft, it is usually much easier to attack a client and to steal the data available to the client than attacking the cloud system, although attacking the cloud may enable access to the data of the entire system.

By attacking a client, an attacker can gain access to all data of the health record for which the user or users of the client are authorized. Clearly, by launching such an attack on a large scale, the amount of stolen data can be increased. Although the potential harm for a health record system is less, it is assumed to be easier to mount attacks against clients. This is due to the fact that security mechanisms (firewalling, intrusion detection and prevention, etc.) by the cloud provider are more sophisticated than those by the weakest link, i.e., home users. However, we note that the compromise of medical data of an individual may result in massive negative consequences for this individual.

On the other hand, if the cloud system is under attack, the entire health record and consequently all the medical data may be compromised, although security is one of the core competences of a cloud provider. A prominent example was the security bug in Google Docs. Google Docs, a cloud based word processing service, had an access-control bug in 2009, which provided users access to documents they had never been granted access for [10]. This fact underpins the argument that even large companies cannot guarantee that their systems are immune from even non-sophisticated attacks.

In context of HRs, one could pose the question: Are such incidents and vulnerability to attacks system-inherent or preventable? We will try to argue that this is not system-inherent in the next section.

7. PRIVACY-RELEVANT REQUIREMENTS AND METHODS

Based on the classification of data in health records, we have discussed potential misuses and the negative impacts on patients' personal surroundings and their well being. Furthermore, we have listed important issues for health records in the cloud; most of them apply to arbitrary services in the cloud. In our opinion, the most critical and an open issue in context of health records is the protection of patient's privacy. This is due to the sensitive nature of medical data and especially actual concepts only cover specific issues [11, 12, 13, 14]. As a consequence, it would be desirable to have a holistic view which integrates all relevant privacy aspects. In providing a first step in this direction, we define, motivate and discuss requirements and their impact on the protection of the patient's privacy as following:

- Unlinkability of patients and their medical data: Any person who even has access to the entire system but is not explicitly granted access rights, cannot link medical data to the patient. This means that a single user is explicitly granted for one action if he/she is the initiator of the action or if he/she obtains a token (to be clarified below) from a granted user to perform the action. For instance, in case of an encrypted document, an explicitly granted user is in possession of the decryption key and no one else can hold this key and decrypt the document. The unlinkability guarantees that it is not possible to determine the links between patients and their medical data. Hence, even if data are

stolen or data leakage as a failure of the cloud provider occurs, medical data cannot be linked uniquely to patients, and consequently patient's privacy cannot be compromised. This first requirement would be counteracted if users and especially patients perform a standard authentication. This is due to the fact that all actions performed could be linked to the authenticated user. Consequently, an insider could easily figure out the relation between administrative and medical data. Thus, we need the two subsequent requirements.

- Non-identification of users: A user, e.g., a patient, browsing his/her health record, can never be identified by any observer of the system (insiders or external parties). This includes the authentication and every subsequent action within the system.
- Unlinkability of actions within the system: Unlinkability means that it is infeasible to link different actions of a user within the system. This aspect in combination with the aforementioned requirement prevents profiling of users. Let us assume that we have linkability of actions within the system but the corresponding patient cannot be identified. By observing the system, it is possible to gradually figure out parts of patient's health records. The more complete the health record obtained, the higher the probability of identifying the patient. Note that if only one part of the health record is known to correspond to a specific patient, this is also true for the entire record. However, by satisfying unlinkability of actions, this information leakage no longer exists.
- Encryption of semi- or unstructured data: Generally, semi- or unstructured data contain information which could be used to identify persons, and thus need to be protected. However, removing this identifying information is non-trivial and usually cannot be performed automatically. Furthermore, there is no guarantee that the metadata contained in these data, e.g., address, gender, relatives, etc., will not be used to uniquely identify the patient (as for instance, suggested by works in the field of k-anonymity [15, 16]). Hence, the only effective way to ensure patient's privacy is to protect documents as a whole. This can be accomplished by means of encrypting these data such that only persons who are explicitly granted are able to decrypt them.
- Independent views of a patient's health record: Patients may define independent views (subsets) of their health records and explicitly grant access to these views. This enables patients to use or present a view of their own health records in specific scenarios. The major purpose is that every user is in possession of a public view, which can be used by other users to integrate data for or share data with this user. Patients can either leave the data in this public view or anonymously move their data to other views. Two additional features of these views are the following: Patients in cooperation with their physicians can define a view for emergency, which can be accessed in case of emergency by physicians. Additionally, patients can define views which they can present in case of enforced disclosure to, for example, current employers or potential employers during job interviews. The patient may choose not to disclose

information concerning his/her cured diseases, such as depression, which are no longer relevant but could negatively influence the decision of the human resource manager. This scenario is not as hypothetical as it seems. In recent years, companies have been discovered to illegally maintain "health records" of their employees. Hence, it is to the user's advantage to be able to prepare views that do not contain potentially compromising information.

In addition to the requirements that are directly connected to the patient's privacy, there are other essential requirements:

- Key management: The number of keys which are managed by the user should be kept to a minimum. Ideally, the user needs only a master key which is stored in a smartcard. All other keys concerning this user are stored encrypted within the HR system and can only be decrypted by the user.
- Key backup: The system needs to provide mechanisms that allow users to backup their master keys for other users or groups of users.
- User-side cryptography: All cryptographic operations are performed by the user's clients. If insiders are taken into consideration, cryptographic operations must not be performed on the server side. For instance, even when using server-based encryption in combination with transmission encryption (e.g. SSL/TLS), the transmitted data will temporarily be available in plaintext at some point of time in the HR system. Specifically, this means that the data will be passed in plaintext from the "transmission encryption" to the component of the HR system which realizes the server-based encryption. Moreover, it must be emphasized that firstly, specific insiders are able to access data in plaintext and secondly, the cryptographic keys used for data encryption is managed in the HR system and consequently is vulnerable to attacks.
- Auditing: All actions within the system, such as authentication, access to data, and their context need to be recorded.

Clearly, in designing such a health record system it is necessary to achieve reasonable efficiency and manageable complexity of the implementation in practice. In this context efficiency means that all operations within a health record, i.e. browsing the health record, integrating, accessing data and searching for specific data, can be performed efficiently. Complexity of implementation means that the costs for implementing methods to achieve all above mentioned requirements should be reasonable. Next, we will focus on methods which can be used to meet these requirements.

7.1. Unlinkability of Patients and Their Medical Data

Unlinkability will be discussed by taking the very simple but meaningful example of a health record illustrated in Figure 1 by means of an Entity Relationship Diagram (ERD). We assume that the system is able to assign every patient an arbitrary number of folders. A typical folder may contain all data related to the general practitioner. The folders are used to structure the health record and can contain structured or semi-structured data (DATA) or unstructured data (DOCUMENT).

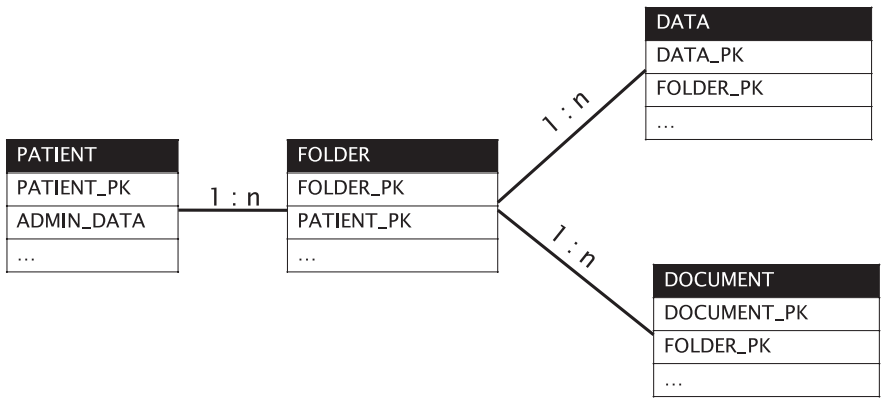


Figure 1. Simplified conceptual model of a health record. PK represents the primary key of a table and also the foreign key in corresponding detail tables. The cardinality 1 : *n* indicates a master-detail relationship between two tables, e.g., one patient can have *n* associated folders.

Before discussing unlinkability in detail, we present some cryptographic preliminaries. We assume the existence of a public-key infrastructure, which means that every user is in possession of a certified public key (PUK) and a private key (SK). We explicitly note that the public-key encryption scheme needs to be secure and thus probabilistic. Hence, RSA-OAEP [17], ElGamal [18] and Cramer-Shoup [19] encryption schemes are suggested. We denote for a given string m the symmetric encryption with key S as $E_S(m)$ and for public key encryption as $E_{PUK}(m)$. The corresponding decryption of a ciphertext c is denoted by $D_S(c)$ and $D_{SK}(c)$ respectively.

To achieve unlinkability between the objects (PATIENT, FOLDER, etc.), pseudonymization of relationships [12, 20, 21] is adopted. In order to pseudonymize a relationship (e.g., PATIENT to FOLDER), the creator of the record chooses a second random primary key per record, encrypted and then integrated into the master table (PATIENT). This second primary key is used as a "foreign-key" in the detail table (FOLDER) in a decrypted way, as shown in Figure 2. It must be mentioned that the primary key and the second primary key are database keys and not cryptographic keys.

Any user who is able to decrypt the second primary key in the master table can easily determine specific details (folders) related to this specific key. For anyone who is not authorized, it is impossible to establish the specific relationship between records of the master table and the detail table. This process of pseudonymization can be easily applied to any hierarchical data model.

In order to allow data sharing, the second primary key is encrypted with a record-specific symmetric key (S) which will be encrypted for all authorized persons using their own public keys (PUK). The resulting tokens containing the primary key of the record (PATIENT_PK), the reference to the grantee (GRANTEE) and the encrypted symmetric key ($E_{PUK}(S)$) are stored in the system. To illustrate this concept, the access

to folders of a specific person is presented below (see Figure 2): First of all, an authorized person, e.g., a physician, requests the record of a particular patient. Using the primary key of this patient (PATIENT_PK), the physician checks whether he/she has a token which references this primary key. By means of his/her private key, the physician decrypts the symmetric key of the token and decrypts the PATIENT_PK_2 of the patient. Finally, the physician uses the PATIENT_PK_2 to retrieve the related folders by means of a query based on the Structured Query Language (SQL). This strategy requires that separate tokens are created for different database tables that need to be pseudonymized. This means that all symmetric keys used to encrypt the second primary key are chosen independently. As a result, for every record and every granted user, a separate token needs to be created.

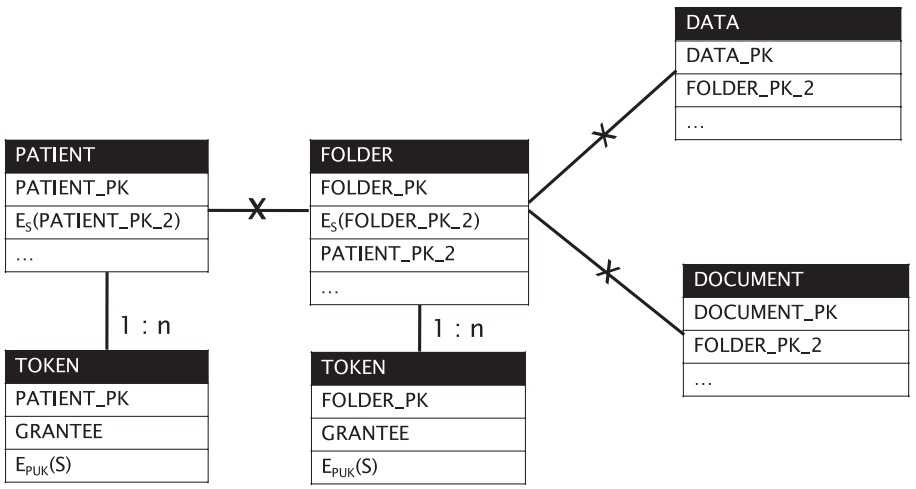


Figure 2. Pseudonymized conceptual model. The conceptual model of Figure 1 with the encrypted second primary keys, e.g., ES(FOLDER_PK_2), and the "foreign keys", e.g., FOLDER_PK_2. The pseudonymization of a relationship is indicated by X. The symmetric key S is encrypted with the patient's public key and stored in a token. Tokens authorize users to gain access to the corresponding data.

Another relevant issue is that users, e.g. physicians, should be able to efficiently find their tokens for specific patients. In the above example, the patient grants *n* physicians (grantees) access to his folders. As a result, a physician must retrieve *n* tokens and to perform a linear search and decrypt every token. If *n* is of moderate size, this approach can be feasible. The advantage of this approach is that the system leaks no information about the identity of the grantees. If this anonymity is not required, integrating the identity of the grantee allows logarithmic searches. Alternatively, through searchable encryption [22], one is able to perform logarithmic searches while maintaining strong privacy guarantees simultaneously.

One important point in this strategy is that the tokens may only be revocated (removed) by the creator of tokens. Therefore, the creator must prove to the health record system that he is authorized to remove a token. Otherwise, anyone could remove any token in the system. Technically, this can be achieved by proof of possession described below. For instance, the creator of a token chooses a random number and encrypts it for the health record system as well as for himself/herself and then adds the two ciphertexts to the token. For removals or updates, the creator decrypts his/her ciphertext and sends the random number confidentially to the health record system. The system will only accept the deletion or update if the provided random number is equal to the number obtained from the decrypted ciphertext of the token. Another issue that should be taken into consideration is that removal of a token does not prevent a user from storing the symmetric key contained in the corresponding token at the client. Hence, the user is able to use the symmetric key although the token was already removed. In the proposed concept, this is true only for structured data since they are not stored encrypted as described in beginning of this section. This can be prevented by a rekeying strategy, i.e., the symmetric key and the second primary key (and the corresponding foreign keys) are replaced by new ones and all related tokens are updated with the new symmetric key. The aforementioned strategy is valid if only the creator of a record (e.g., in table FOLDER) provides tokens to other users. However, if these granted users create tokens for other users, e.g., corresponding to the aforementioned record, then the strategy needs to be modified. The basic idea is that these "forwarded" tokens contain a reference to a token that was initially provided by the creator of the record. More precisely, a "forwarded" token includes a symmetric key to decrypt the content of the initial token and, consequently, the rekeying strategy does not affect the "forwarded" token. In our concept, rekeying should only be performed by the creator of an initial token. As a consequence, users who want to revoke "forwarded" tokens need to notify the creator before rekeying.

The pseudonymization of relationships presented above enables that all records of a table, e.g., FOLDER, can be partitioned according to the "foreign key" as illustrated in Figure 3. However, this does not imply that one can find a link to any other record in other tables, e.g., the patient. By applying simple obfuscation techniques as discussed in [20], one can hide the actual size of the sets in the partition and thus reduce the information accessible to an adversary.

Performance loss can be kept to a minimum when the access to the health record is top down (from the patient to the data). In this case, only two additional decryptions per level (table access) are necessary.

PATIENT	
PATIENT_PK	E_S(PATIENT_PK_2)
101	E_S(201)
102	E_S(202)
103	E_S(203)
104	E_S(204)
…	

FOLDER		
FOLDER_PK	E_S(FOLDER_PK_2)	PATIENT_PK_2
301	E_S(401)	201
302	E_S(402)	201
303	E_S(402)	201
304	E_S(403)	202
305	E_S(404)	202
306	E_S(405)	202
307	E_S(406)	203
308	E_S(407)	203
309	E_S(408)	203
…		

Figure 3. Pseudonymized relationship between patients and folders. Folders with the same PATIENT_PK_2 in table FOLDER, e.g. 201, are related to a unique but unknown person. Only users who are able to decrypt ES(PATIENT_PK_2) using the user specific symmetric key S in table PATIENT are able to identify the person.

7.2. Non-Identification of Users and Unlinkability of Actions within the System

A first approach to guarantee that a user cannot be identified by the health record system is to employ anonymous authentication protocols. These protocols can authenticate users of a system without revealing user's identity. However, this entails problems in context of the Internet, since messages transferred over the Internet contain explicit addresses, i.e., IP-addresses, which may be used to identify the user. Hence, the user needs to communicate with the health record system via an anonymous communication channel [23]. Since every action within the system needs to be authorized, an initial but very inefficient solution would be to use a single anonymous authentication for every action. This is problematic because the anonymity sets (the groups used for anonymous authentication) and the access to user-related data increase the probability to identify the authenticated user. Both aforementioned problems can be solved by using anonymous yet authorized transactions. This is discussed in more details in the following section.

7.2.1. Anonymous Authentication

In contrast to conventional authentication methods, which establish a unique identification of the authenticating user, anonymous authentication enables users to authenticate without disclosing their own identity to the verifying party. There are many different techniques for anonymous authentication in the literature [24, 25, 26, 27]. In general, a user in an anonymous authentication protocol is able to prove membership of a group of authorized users to a verifier (the health record), whereas the verifier does not obtain any information on the actual identity of the authenticating user. Using the aforementioned public-key infrastructure, anonymous authentication from public-key encryption as proposed in [27] can be employed efficiently with moderate

implementation effort. Loosely spoken, this can be realized as a parallelization of a challenge-response protocol based on public-key encryption. The system encrypts a randomly chosen string using the public keys of a set of users (the anonymity set) and sends this sequence of public keys to the anonymous user. If the anonymous user is able to decrypt one of these ciphertexts and provides the same strings as the one sent by the system, the system is guaranteed that the user knows one of the private keys corresponding to the users in the anonymity set. However, the system cannot determine the exact identity of the user and the user is anonymous within the anonymity set.

7.2.2. Anonymous Communication

Mechanisms that provide anonymity and unlinkability of messages sent over a communication channel are denoted as anonymous communication techniques and have been studied intensively in recent years [23]. There are several implementations available which are realized as overlay networks and can be used for Internet communication without any implementation effort. These anonymous communication channels provide anonymity of users against eavesdroppers and curious communication partners who are no longer able to identify users by their messages. Anonymity, especially against curious communication partners, can be preserved if electronic interaction does not rely on additional identifying information at higher network layers, i.e., the application layer. For example, a user who queries a public web page using an anonymous communication channel may remove all identifying information from higher network layers, and thus will stay anonymous. The efficiency of this method depends highly on the degree of anonymity, e.g., the number of mix nodes, but it implies latencies in the communication. We note that in institutional access (such as access of all users in a hospital) to health records, anonymous communication can be implemented more efficiently by dedicated proxies.

7.2.3. Anonymous yet Authorized Transactions

When single actions within the health record system need to be unlinkable, an initial but very inefficient approach would be to perform independent anonymous authentications for every single action. A far more efficient way to achieve unlinkability is to combine anonymous authentication with unlinkable transactions as proposed in [28] (see Figure 4). After an initial anonymous authentication, the user receives a ticket which can be linked to neither the authentication nor the user, but can be used to authorize a single action in the system. Every showing of a ticket prompts issuing a new ticket, whereas this ticket can be shown in a way that it is unlinkable to either the user or its issuing.

Combination of anonymous communication, anonymous authentication and anonymous yet authorized transactions allows users to work anonymously but with authorization and unlinkability in the HR system.

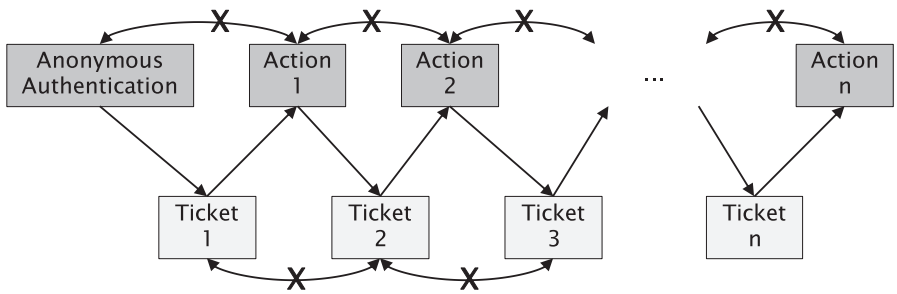


Figure 4. Unlinkable actions and tickets. After an anonymous authentication, a ticket is issued by the HR system. A ticket authorizes one action within the system, and showing a ticket results in issuing a new one. Consequently, issuing and showing of tickets are unlinkable, and so are actions (indicated by X).

7.3. Encryption of Semi-structured or Unstructured Data

To encrypt a document (semi-structured or unstructured data), the owner of the document randomly generates a symmetric content encryption key, encrypts the document and integrates the document into the system. The user then generates for every explicitly granted person a token. This token contains the symmetric key and meta-information regarding the document, and is encrypted with the person's public key. In order to efficiently find tokens in the system, a reference to the document, e.g., DOCUMENT_PK, must be added. Search for tokens and the revocation of tokens can be performed in analogy to the approach discussed in Section 7.1.

7.4. Independence of a Patient's Health Record

With regard to the technical aspects concerning the realization of views, the following requirements are defined:

- Each user can define any number of independent views, with each of them containing a subset of the user's health record.
- Insiders are not able to link views to the user.
- When a user opens a view, no information about the existence of other views of this user is revealed. This is necessary, for instance, in the situation of an enforced disclosure of the view, where the user presenting the "non-compromising" view can plausibly deny the existence of other views.

The first requirement above can be achieved by modifying the conceptual model in Figure 1, where an additional entity PATIENT_VIEWS needs to be integrated between PATIENT and FOLDER as illustrated in Figure 5.

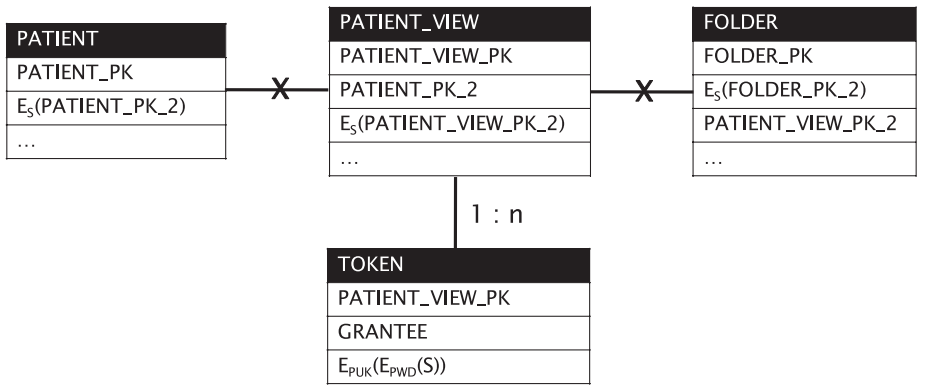


Figure 5. Conceptual model of independent views. A user can have several views containing different subsets of his/her health record. The relationship between patients and views is pseudonymized. For every view, a token is created which contains an encrypted symmetric key S to reconstruct the relationship between PATIENT_VIEW and FOLDER. This ciphertext is further encrypted with the user's public key such that insiders cannot gain information on activated views.

In order to prevent insider attacks, the relationships between PATIENT and PATIENT_VIEWS as well as PATIENT_VIEWS and FOLDER are pseudonymized. From this point on, it can be assumed that the user has successfully conducted an anonymous authentication and has decrypted the second primary key which is stored encrypted in the PATIENT table. This second primary key can be used to retrieve all potential views of the user. We use the term "potential" because initially the user created a number of deactivated views. In order to activate a view, the user chooses a password (PWD, see Figure 5) to encrypt a random symmetric key. This symmetric key S is used to encrypt the second primary key of the table PATIENT_VIEW similar to that discussed in Section 7.1. The resulting token, containing the encrypted symmetric key and the reference to the user's view, is stored in the system. In order to prevent ungranted access of information by insiders, it is necessary to initially generate a random dummy token for every deactivated view. Furthermore, it is essential that activating one view implies that all tokens of the remaining views need to be modified. Hence, we suggest further encrypting the ciphertext resulting from encrypting the symmetric key with the password using the user's public key. Note that the used public key encryption scheme is probabilistic. Such approach enables efficient re-encryption (modification) of the ciphertexts without the need of entering all passwords corresponding to activated views. Figure 6 illustrates the use of independent views from a patient's point of view.

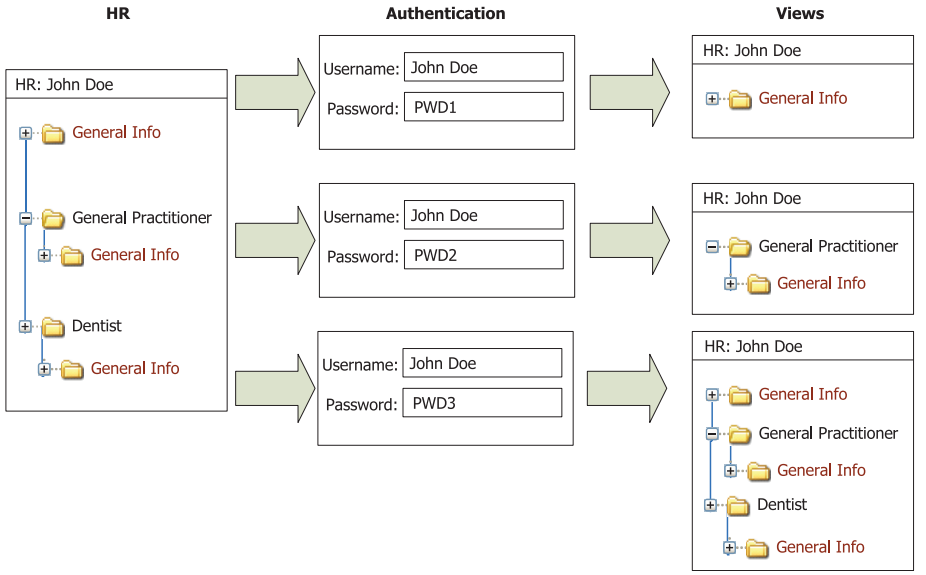


Figure 6. Independent views of a patient's health record. A hypothetical user has three active views. By entering PWD1, PWD2, or PWD3, the respective view is presented to the user.

7.5. Key Management and Backup

In the proposed concept, each patient holds a public/private key pair and one password for every activated view of his/her health record. Such key pair is used for anonymous authentication. The public key is also used during the creation of tokens to integrate information, e.g., the granted person, symmetric encryption key, second primary key, which can be decrypted by the receiver using the private key. Storing the private key solely on a smartcard is not recommended, since private keys can be not exported from smartcards. Consequently, key backup is not possible and loss of a user's card would create serious problems. Therefore, the smartcard should only be used to "unlock" the private key, which is stored encrypted within the system and the encryption key may be shared among other users through secret sharing. The aforementioned passwords are used to open activated views. It is important that they are chosen independently of each other in order to prevent attacks based on knowing one password and the strategy to derive all other passwords. Furthermore, they should be chosen according to a strong password policy. The access to the private key and the passwords is essential in our model. Loss of such information makes the patient no longer have access to his/her health record. Therefore, a proper key backup strategy is imperative. In our model, we propose the following two approaches to guarantee the availability of the patient's private key or passwords. The first one is the sharing of a patient's key with certain other patients, e.g., relatives, to be accomplished by means of tokens. The second approach is the sharing of a patient's key among a specific group so that certain subsets of this group are able to reconstruct the key. This can be accomplished by means of

threshold secret sharing schemes [29] or secret sharing schemes for general access structures [30]. This approach forgoes the need that every member of the group is fully trustworthy.

7.6. User Side Cryptography

In the cloud computing paradigm, a web browser is usually used as a client in order to avoid problems associated with the distribution of the client software. In this case, client side cryptography cannot be achieved solely by script languages (e.g., JavaScript), especially when accessing local resources such as smartcards. Therefore, it is necessary either to change the paradigm of web browsers, i.e., to include a standardized and cross-platform Application Programming Interface (API), or to use alternative concepts like Java Applets or Microsoft's Silverlight (most likely with proprietary cryptographic libraries). Nevertheless, in order to avoid the client software to act as a Trojan horse, it is crucial to certify and digitally sign the code by a third party. Otherwise, the client could illicitly leak out cryptographic key material.

7.7. Auditing

Auditing in context of anonymous authentication is basically pointless because users cannot be uniquely identified. The only way that this is compliant with our paradigm is that users are solely able to audit actions, e.g., access operations on their own health records. To solve this problem, it is necessary to consider the following scenario. Accessing data can solely be executed by using tokens. Consequently, the system is able to log access information for these tokens (a reference of the token, type of action and the time). To link this token to the granted person, the creator of the token integrates the identity of this person via encryption. If the encryption is executed by means of the pubic key of the patient, the patient is able to identify the accessing persons.

8. CONCLUSION

This paper addresses privacy issues in electronic health records and personal health records in context of the emerging field of cloud computing. Privacy issues are of great importance in dealing with sensitive medical data, and are however of secondary priority in current cloud applications. Necessary technical measures have not been broadly discussed so far and are not available in common solutions. We have focused particularly on insider attacks because the prevention against them automatically covers other attacks. This is true because external adversaries can at most obtain the privileges of an insider when breaking into the system, or if data are stolen or leaked, intruders having access to these data can act like insiders. To protect patient's privacy against potential insiders' attacks, we have combined a set of non-standard methods which have been adjusted accordingly. Nevertheless, a cloud provider is still able to host an efficient health record system without being able to link patients and their health records.

ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the presentation of the paper.

REFERENCES

[1] Google.com. Google Health - Personal Health Record Service. www.google.com/health.

[2] Microsoft. Microsoft HealthVault. www.healthvault.com.

[3] Computer Security Institute (CSI). Computer Crime and Security Survey 2007. http://www.gocsi.com/forms/csi survey.jhtml

[4] Gain, B. Cloud Computing & SaaS In 2010 - What To Expect After The Uncertainty & Hype Fade. Processor 32(1), 2010.

[5] Rittinghouse, J., and Ransome, F. Cloud Computing: Implementation, Management and Security. CRC Press, 2010.

[6] Mell, P., and Grance, T. The NIST Definition of Cloud Computing. National Institute of Standards and Technology, http://csrc.nist.gov/groups/SNS/cloud-computing

[7] Tang, P.C., Ash, J.S., Bates, D.W., Overhage, J.M., and Sands, D.Z. Personal Health Records: Definitions, Benefits, and Strategies for Overcoming Barriers to Adoption. J Am Med Inform Assoc. 13 (2): 121-126, 2006.

[8] Frick, U., Baer, N. et al. Chronisch Kranke einstellen? Eine experimentelle Vignettenstudie unter Personalmanagern. In Proc. FFH 2008, pages 45-50, Martin Meidenbauer Verlag, 2008.

[9] Jensen, M., Schwenk, S., Gruschka, N., and Lo Iacono, L. On Technical Security Issues in Cloud Computing. In Proc. of the IEEE International Conference on Cloud Computing, pages 109-116. IEEE, 2009.

[10] Thomson, I. Google Docs leaks private user data online. http://www.v3.co.uk/vnunet/news/2238122/google-docs-leaks-private.

[11] Riedl, B., Grascher, V., and Neubauer, T. A Secure e-Health Architecture based on the Appliance of Pseudonymization. Journal of Software 3(2): 23-32, 2008.

[12] Caumanns, J. Der Patient bleibt Herr seiner Daten. Realisierung des eGK-Berechtigungs-konzepts über ein ticketbasiertes, virtuelles Dateisystem. Informatik-Spektrum 29 (5): 323-331, 2006.

[13] Alhaqbani, B., Fidge, C. Privacy-Preserving Electronic Health Record Linkage Using Pseudonym Identifiers. In Proc. of HealthComm 2008, pages 108 – 117, IEEE Communications Society, 2008.

[14] Huda, N., Sonehara, N., and Yamada, S. A Privacy Management Architecture for Patient-Controlled Personal Health Record System. Journal of Engineering Science and Technology 4(2): 154-170, 2009.

[15] Sweeney, L. k-Anonymity: a Model for Protecting Privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10(5): 557–570, 2002.

[16] Samarati, P. Protecting Respondents' Identities in Microdata Release. IEEE Trans. Knowl. Data Eng. 13(6): 1010–1027, 2001.

[17] Bellare, M., and Rogaway, P.: Optimal Asymmetric Encryption. In Proc. of EUROCRYPT 1994. LNCS vol. 950, pages 92-111. Springer-Verlag, 1994.

[18] El Gamal, T.: A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In Proc. of CRYPTO 1984, LNCS vol. 196, pages 10-18. Springer-Verlag, 1984.

[19] Cramer, R., Shoup, V.: A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack. In Proc. of CRYPTO 1998, LNCS vol. 1462, pages 13-25. Springer-Verlag 1998.

[20] Stingl, C. and Slamanig, D. Privacy-enhancing methods for e-health applications: how to prevent statistical analyses and attacks. Int. J. Bus. Intell. Data Min. 3(3): 236-254, 2008.

[21] Slamanig, D., and Stingl, C. Privacy Aspects of eHealth. In Proc. of ARES 2008, pages 1226-1233. IEEE Computer Society, 2008.

[22] Bellare, M., Boldyreva, A., O'Neill, A. Deterministic and Efficiently Searchable Encryption. In Proc. of CRYPTO 2007. LNCS, vol. 4622, pages 535-552. Springer-Verlag, 2007.

[23] Danezis, G. and Diaz, C. A Survey of Anonymous Communication Channels. Technical Report MSRTR-2008-35, Microsoft Research, 2008.

[24] Ateniese, G., Camenisch, J., Joye, M., Tsudik, G. A Practical and Provably Secure Coalition-Resistant Group Signature Scheme. In Proc. of CRYPTO 2000. LNCS, vol. 1880, pages 255-270. Springer-Verlag, 2000.

[25] Boneh, D., Boyen, X., Shacham, H. Short Group Signatures. In Proc. of CRYPTO 2004. LNCS, vol. 3152, pages 41-55. Springer-Verlag, 2004.

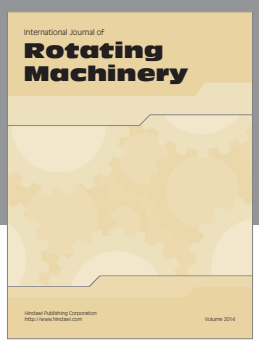
[26] Teranishi, I., Sako, K. k-Times Anonymous Authentication with a Constant Proving Cost. In Proc. of Public-Key Cryptography 2006. LNCS, vol. 3958, pages 525-542. Springer-Verlag, 2006.

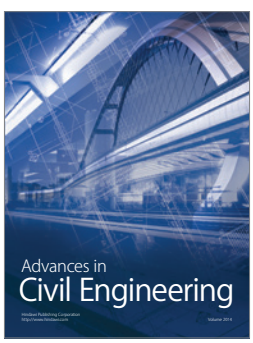
[27] Slamanig, D., Schartner, P., Stingl, C. Practical Traceable Anonymous Identification. In Proc. of SECRYPT 2009, pages 225-232. INSTICC Press, 2009.

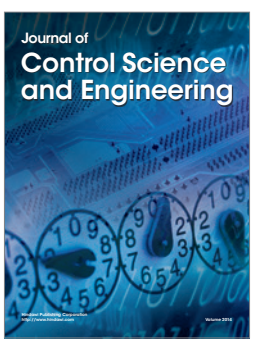
[28] Slamanig, D., and Rass, S. Anonymous But Authorized Transactions Supporting Selective Traceability. In Proc. of SECRYPT 2010, IEEE Communications Society, 2010.

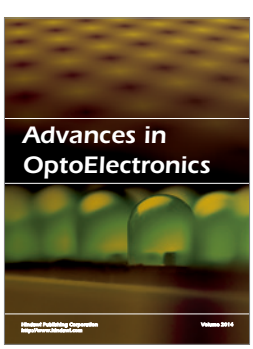
[29] Shamir, A. How to Share a Secret. Commun. ACM 22(11): 612–613, 1979.

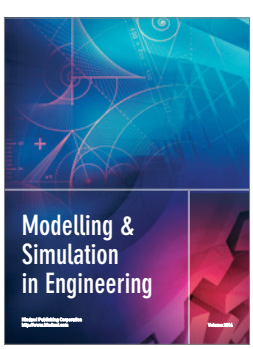
[30] Benaloh, J.C., and Leichter, J. Generalized Secret Sharing and Monotone Functions. In Proc. of CRYPTO 1988. LNCS, vol. 403, pages 27-35, Springer-Verlag, 1988.

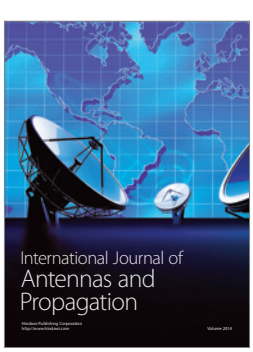


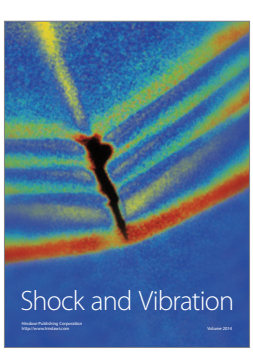


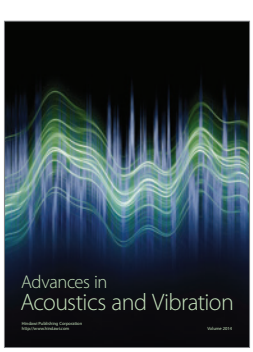












Hindawi

Submit your manuscripts at
http://www.hindawi.com