A Decentralized Pseudonym Scheme for Cloud-based eHealth Systems

Liangyu Xu and Armin B. Cremers
Institute of Computer Science, University of Bonn, 53117 Bonn, Germany

Keywords: Pseudonym, EHR, Healthcare, Cloud, eHealth, Security, Privacy.

Abstract: A decentralized pseudonym scheme is proposed for providing storage, encryption and authentication of patients' EHRs in cloud-based eHealth systems. The pseudonyms of a patient are generated from the patient's secrets and each of them is used as the index of an EHR entry of the patient. An encryption key derived from the pseudonym can be used to encrypt the corresponding EHR entry. The pseudonyms can also be used for the patient proving the ownership of the EHR without disclosing the identity of the patient. Some protocols and remarks for using the pseudonym scheme are also discussed.

1 INTRODUCTION

Storing shared EHRs (Electronic Healthcare Record) (Garets & Davis, 2006) at a place which can be accessed from anywhere is an attractive feature in eHealth (electronic healthcare) systems. Each patient owns an EHR containing many EHR entries which detail the healthcare information of the patient. The sharing of EHR can improve the quality of diagnosis and treatment, and even enable the patients to view and manage their own EHRs through their own devices (Ruland et al., 2008).

The cloud (Mell & Grance, 2011) is an ideal media for storing EHRs providing wide access, because it is able to offer ubiquitous services to customers over the Internet (Rui & Ling, 2010). However, many challenges exist for adopting the cloud as the storage media for EHR data in an eHealth system. Since the EHRs of many patients are stored at a central place, the cloud, there must be an efficient approach to index the EHR entries and map the EHR entries to their owners. i.e., the cloud needs to be able to provide fast access to the EHR entries from queries of different patients. Another challenge is that since the cloud's resources are assumed to be publicly accessed and the cloud itself may also be malicious (Deng et al., 2011), the storage media for the patients' EHRs is not physically secure anymore, which makes the EHRs face more potential attacks from external adversaries as well as from corrupt insiders. Thus, the contents of the EHRs in the cloud should be properly protected, usually by encryption, and should only be accessible by the appropriate users, i.e. the patients themselves and some authorized persons. Another challenge is that the patients' identities should not be disclosed and/or linked to the EHRs by any person not intended to be able to. Even the cloud should not know the identities of the patients. Although the EHR contents are encrypted, it is not secure to expose the patients' identities, for attackers will possibly acquire general knowledge of the patients' healthcare activities (e.g., the attackers may discover at least which doctors one patient has visited from the doctors' signatures) and even more from analyzing the contents in EHRs stored in the cloud (Stingl & Slamanig, 2008).

However, sometimes the EHR entries need to be updated by the patients, doctors or pharmacists (e.g., an update of a prescription in an EHR entry when the prescription is used in a pharmacy; the patient manages his own EHR entries like setting custom access control; etc.), so the cloud has to be able to authenticate the patients and verify the ownership of the EHR entries in order to avoid illegal changes by attackers. It would be better that the authentication process not leak any information about the patients' identities to the cloud or possible attackers.

In this paper we discuss how a decentralized pseudonym scheme can act in the cloud-based eHealth systems for the purpose of EHR storing and protecting, while providing the ability for the cloud to authenticate the ownership of EHRs without

Xu L. and Cremers A..
A Decentralized Pseudonym Scheme for Cloud-based eHealth Systems.
DOI: 10.5220/0004747602300237
In *Proceedings of the International Conference on Health Informatics* (HEALTHINF-2014), pages 230-237
ISBN: 978-989-758-010-9

knowing the identities of the patients, besides protecting the identities of patients.

The remainder of this paper will be organized as follows. We list and discuss some previous related work in section 2. Section 3 gives a simple model of cloud-based eHealth systems and shows generally how pseudonyms are used to protect patient privacy while performing EHR storing, retrieving, managing, and ownership authenticating. A concrete pseudonym design will then be presented in section 4. Several detailed evaluations and points of discussion will be addressed in section 5. The last section 6 presents some conclusions and an outlook on future work.

2 RELATED WORK

Many efforts have been made recently to migrate EHRs which are scattered in clinics and hospitals into “the cloud”; an important issue of this task is security and privacy concerns, because the disclosure of healthcare information of patients may cause severe problems, especially to the patients (Alemán et al. 2013). There exist some healthcare clouds such as Microsoft HealthVault (Microsoft, 2007), where registered patients claim and manage their personal EHRs themselves, including sharing their EHRs with selected doctors or other desired persons (These EHRs are often referred as PHRs (Tang et al., 2006)). The privacy protection of such kind of services depends on the security mechanism implemented by the cloud (Löhr et al., 2010). A prerequisite to guarantee the privacy in such services is that these healthcare cloud providers should be completely trustworthy to protect the EHRs properly. Other research proposes solutions that remove the high dependency on trusting the cloud, cf. e.g. (Alhaqbani & Fidge, 2008; Li et al., 2011). The EHRs in these systems are encrypted by keys unknown to the cloud. These solutions may protect the patients’ privacy better, but they make the systems considerably more complex.

To protect the identities of the patients, pseudonyms (Pfitzmann & Köhntopp, 2001) can be utilized to hide the real identity information. Patients’ real identities appearing in the EHRs are removed or replaced by the pseudonyms. There are some pseudonym schemes in which a centralized party (which needs to be trusted) generates the pseudonyms for the patients (Alhaqbani & Fidge, 2008). In such schemes, it is easy to avoid pseudonym collision, and the centralized party can easily provide the service for other parties (e.g. the cloud) of authenticating the patients. The centralized party has to be online with high availability. Moreover, the existence of such a centralized party will possibly harm the privacy of the patients, as a successful attack on the centralized party or a corrupt insider will cause privacy infringement to the patients. In eHealth systems, as the EHR data are top private information for patients, the existence of a trusted party that has the ability to know patients’ identities or the plaintext of their EHR data and has potential risk to be attacked would not be accepted by many patients.

It is highly desirable that pseudonym generation be decentralized, where no trusted party exists for generating and authenticating pseudonyms. There are some decentralized pseudonym schemes for eHealth systems. In (Li et al., 2011), such a pseudonym scheme is used to index the EHR entries. The pseudonyms can only be reproduced to retrieve the EHR entries from the cloud by fetching two parts of secret information which were separately stored in two trusted parties when the pseudonyms were generated at the first time. However, this solution would require the patients to store much information in the smart cards, and the existence of trusted parties places disclosure risk on the patients’ privacy. In (Lysyanskaya et al., 2000), a general pseudonym scheme is proposed for generating pseudonyms for one user (possessing a secret) and using the credentials issued by one organization at other organizations without revealing the user’s identity. Each user’s pseudonym is generated based on the user’s secret with the organization’s participation. They minimize the dependency on a trust party, and the security relies on the user-only known secrets. However, the scheme might encounter problems when a central cloud is utilized in eHealth systems.

In a cloud-based eHealth system, Not only should a pseudonym scheme be carefully designed to protect the identities of the patients from being disclosed, but also some protocols are needed to apply the pseudonym scheme correctly to protect the privacy of the patients while keeping the healthcare processes running smoothly. One motivation of this paper is to design such a pseudonym scheme which can pseudonymize patients’ identities with provable security, and also provide corresponding encryption key management and authentication algorithm without disclosing patient’s identities. Another intention of the paper is to apply the pseudonym scheme to a typical cloud-based eHealth system, by discussing the possible risks of privacy and identity disclosure and corresponding solutions.

3 USE OF PSEUDONYMS IN CLOUD-BASED eHEALTH SYSTEMS

3.1 Our Model of a Cloud-based eHealth System

We model a basic cloud-based eHealth system as shown in Figure 1. The cloud plays the role of associating almost all the participating eHealth entities such as patients, doctors, pharmacists, and health insurance companies (will be abbreviated as insurance companies in the following). We note that the pseudonym scheme proposed in this paper does not depend on the existence of insurance companies. It would be easier to adapt the pseudonym scheme in eHealth systems without an insurance company as the insurance companies could also be possibly curious to know the health information of their clients (i.e. patients).

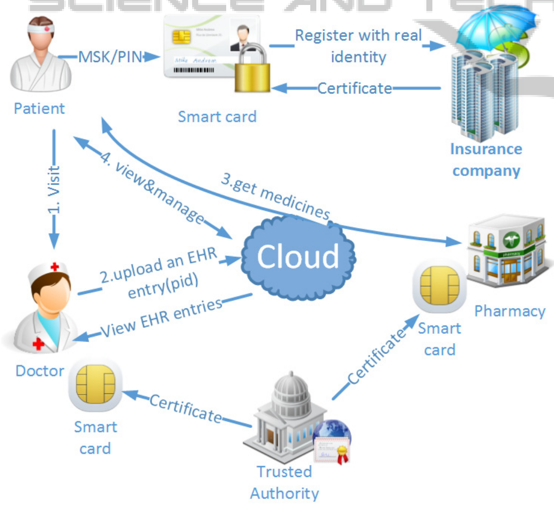

Figure 1: A simple model of cloud eHealth systems.

Each patient has a set of secrets consisting of MSK and PIN, which are only known by the patient himself, where MSK is the main secret key of the patient, usually stored in the protected memory of a smart card, for which a PIN is employed to authenticate the patient to avoid abuse. The MSK of the patient will be used to generate pseudonyms, encrypt the EHR data, prove the patient's ownership of the EHR to the cloud and in some other functions. Hence, it should be properly protected in the smart card and safely backed up by the patient in case of loss of the smart card. The special issues of smart card loss are discussed in section 5.

The patient's real identity is only known by the insurance company while the patient has to register there in the beginning. Each patient will get a certificate from the insurance company with a corresponding private key which is only known by the patient (We will abbreviate the patient's corresponding private key from insurance company as SKI). The patient's identity should not be included in the certificate. Typically, a certificate number, which can only be mapped to the patient's real identity by the insurance company, a token of the insurance company, the public key of the patient, and the period of validity will be necessary in the patient's certificate.

The healthcare providers (e.g. doctors and pharmacists) also need to get certificates from a trusted authority with self-known private keys. The trusted authority could be the health department of the government which confirms the healthcare providers' qualifications by issuing them the digital certificates in which information like healthcare categories and healthcare providers' identities can be enclosed. The certificates and the corresponding private keys could also be stored in the protected memory of smart cards issued to the healthcare providers with authenticating PINs. The healthcare providers' private keys can be used to sign the record sections of the patient's EHR data and prescriptions. The private keys of the patients (SKIs) and the healthcare providers would be used for signing bills, which can be received through the cloud and will be paid by the insurance company. The billing procedure is not depicted in the figure for simplicity.

3.2 Use of PID (Pseudonyms) to Conceal Patients' Identities

A typical pseudonym generation occurs at a doctor's practice when a patient visits the doctor, taking along his smart card. The doctor checks the validity of the patient by checking certificate and challenging the patient's SKI, with the insurance company's root certificate which could also be issued by a trusted authority. The doctor does not necessarily know the identity of the patient. But in practice, as the doctor is assumed to be trusted, it is usually the case that the identity (at least parts of identity information such as name, gender, age, and telephone number) of the patient is known to the doctor. After the doctor's diagnosing and treating, the doctor writes down a record (including the diagnosis, examinations, treatment and other information), with a prescription stating the medicines that the patient needs to take and buy at a pharmacy. Meanwhile, a new pseudonym (PID) and an encryption key (EK) are generated by the patient's smart card from the MSK. Both the record and the prescription will be encrypted by the encryption key and signed by the doctor's private key. Then, all the data from the doctor is uploaded to the cloud along with an index header, the new PID, forming an EHR entry for the patient. Thus, an EHR entry needs to be at least compatible with the following sections where "||" means concatenation,

EHR entry = PID || Enc_{EK}(record) || sig_d(record) ||Enc_{EK}(prescription) ||sig_d(prescription).

The cloud can validate the doctors by verifying their certificates and challenging their private keys with the trusted authority's help to avoid illegal uploading of EHR entries. Actually, no one else is able to generate a meaningful EHR entry for a patient, as nobody can generate valid EKs or PIDs of the patient without knowing the MSK and each pair of valid PID and EK is only used for one EHR entry. Forged EHR entries can be easily recognized and removed by the patient through checking the correctness of decryption or the PIDs.

3.3 Use of PID to Index EHR Entries

The cloud stores the EHR entries of each patient by indexing the PID section of the EHR entries. We expect that each EHR entry would have a unique PID. The special problem of PID collision is discussed in section 5. The cloud does not keep any registration information of the patient, so the cloud does not know the identity of the patient other than the PIDs. As most of the essential contents in the EHR entries are encrypted with the encryption keys which are unknown to the cloud, it is unable to disclose the identities of patients or any meaningful information in the EHR data, even if the cloud is curious to know. For the same reason, attackers who can access all EHRs in the cloud can neither get any useful information from the EHRs nor the identities of the patients.

The cloud returns the EHR entries upon the PIDs the patients reproduce and send to the cloud when the EHR entries are retrieved from the cloud.

3.4 Use of PID to Authenticate the Ownership of EHR

Pseudonyms are used by the cloud to authenticate the patients' ownerships on their EHRs when the EHRs need to be updated by their owners. A typical scenario for an EHR entry that needs to be updated is when a patient goes to a pharmacy to buy medicines by using the prescription which is enclosed in an EHR entry. The patient retrieves the EHR entry from the cloud by providing the cloud a PID (the PID of the EHR entry with an unused prescription can be temporally stored in the smart card for fast query) and shows the decrypted prescription with the doctor's signature to the pharmacist. The pharmacist can check the prescription's signature of the doctor. If the signature is valid, the pharmacist generates an additional signature on the prescription indicating that the prescription has been used and asks the patient to update the original prescription's signature section of the EHR entry by including the pharmacist's signature. The cloud needs to check beforehand whether the patient is the owner of the EHR entry or not. Only if he is, the cloud updates the old prescription's signature section by adding the pharmacist's signature. After the pharmacist confirms that the patient has updated the prescription signature, the medicines are sold to the patient. Due to the updating of the prescription signature, the updated EHR entry with a pharmacist's signature looks like this,

updated EHR entry = PID || Enc_{EK}(record) || sig_d(record) ||Enc_{EK}(prescription) ||sig_d(prescription) || sig_p(prescription).

4 THE DESIGN OF A NEW PSEUDONYM SCHEME

4.1 Secrets Setup

A patient's main secret key MSK and PIN are initially set and only known by the patient when the patient receives a blank smart-card from a system provider that may also deploy the healthcare application software in the cloud. The initialization of MSK and PIN needs the help of some software coupling with the smart card. The MSK is stored in the protected memory of the patient's smart card in which the password PIN usually can be set by the patient to avoid abuse of the smart card.

The patient chooses a k-bits (k is usually no less than 160) prime integer q, and another prime number p (the size of p is usually not less than 512 bits) which satisfy $q|(p\text{-}1)$. By Z_p^* we denote a multiplicative group modulo p. The patient finds $g \in Z_p^*$, to be of order q modulo p. Then g is the generator of the subgroup G_q . By randomly choosing$x \in G_q$, a patient's MSK is formed: [x, g, p,

q].

According to difficulty of the discrete logarithm problem (McCurley, 1990), given $g,\ h \in G_q$, such that h was selected from G_q uniformly at random, it is hard to compute an integer x such that

$$g^x = h \bmod p.$$

The complexity to find such x is no less than O($\sqrt{q}$) (Lim & Lee, 1997). For ease of notation, we will sometimes drop the "mod p" part of the arithmetic expressions in G_q . We build up our secure pseudonym scheme based on the discrete logarithm problem in the subgroup G_q.

4.2 Algorithm for Generating PIDs

Assume that previously a patient has already used i PIDs (i.e., PID_1, PID_2, ···, PID_i). We express PID_0 as (a_0, b_0) which is (0, 0) by default (refer to section 5 for more considerations of choosing PID_0 to avoid pseudonym collisions). Following is the algorithm to generate the i+1th pseudonym PID_{i+1} where "||" means bit concatenation.

```
INPUT: $PID_0 = (a_0, b_0)$, $i$, MSK, PIN (to be authenticated by the smart card)
OUTPUT: $PID_{i+1}$, $EK_{i+1}$
$EK_{i+1} = KHash\ (i+1||a_0||b_0,\ x)$, where $KHash$ is a keyed hash function by key $x$
$a_{i+1} = g^{EK_{i+1}+Hash(a_0||b_0) \bmod q}$, $b_{i+1} = a_{i+1}^x$
$PID_{i+1} = Hash(a_{i+1}||b_{i+1})$
```

Since the order number of the last used PID is needed to generate the next new PID, the order number of the last PID should be stored somewhere, e.g., in the smart card. EK_{i+1} can be used as the encryption key (it might be necessary to be truncated or padded according to the encryption algorithm) for encrypting the private contents of the EHR entry with the index of the new pseudonym PID_{i+1}.

4.3 Algorithm for Reproducing PIDs

All the PIDs and EKs can be reproduced one by one through the following algorithm.

```
INPUT: $PID_0 = (a_0, b_0)$ , MSK, PIN, the number $last$ of last used pseudonym $PID_{last}$
OUTPUT: $PID_1 \sim PID_{last}$, $EK_1 \sim EK_{last}$
FOR i=1 to last DO
    $EK_i = KHash(i||a_0||b_0, x)$,
    $a_i = g^{EK_i + Hash(a_0||b_0) \bmod q}$, $b_i = a_i^x$
    $PID_i = Hash(a_i||b_i)$
```

The above algorithm shows how to reproduce all the used pseudonyms and the encryption keys. In fact any single PID_i and EK_i pair can be reproduced by executing one single loop of the above algorithm.

4.4 Protocol for Authenticating the Ownership of EHR Entries

Following is the protocol for the cloud (abbreviated as C) verifying a patient's (abbreviated as P) ownership of one EHR entry with pseudonym PID, where "→" means sending a message.

```
INPUT: an EHR entry with $PID$ in the cloud; $p$, $q$ of the patient's MSK are known by the cloud
OUTPUT: Yes or No.
P → C: Patient sends ($a$, $b$) such that $PID = Hash(a||b)$
C:     checks whether $PID$ ?= $Hash(a||b)$. If not, C returns No; otherwise continues.
P → C: Patient randomly chooses $s$, calculates and sends ($A$=$a$, $B$=$a^s$ mod $p$)
C → P: Cloud randomly chooses and sends $c$
P → C: Patient computes and sends $y = s+cx$ mod $q$
C:     checks $a^y$ ?= $Bb^c$ mod $p$. If yes, C returns Yes; otherwise returns No.
```

The security of the above algorithm and protocol relies on the difficulty of the discrete logarithm problem in G_q and the one-way property of the hash function. Only the patient possessing the secret x can respond to the cloud's challenge correctly. Anybody else who wants to impersonate the patient is required to compute the discrete logarithm in G_q which is conjectured to need the complexity of O($\sqrt{q}$) (Lim & Lee, 1997).

5 DISCUSSION

5.1 Risks and Countermeasures in Protecting Patients' Privacy by Pseudonyms

5.1.1 Cloud's Knowledge of Complete PID Sets of Patients

A patient's frequent access to his EHR may let the cloud know all the patient's PIDs. Since all the patient's pseudonyms are independent without knowing the patient's secrets, common attackers could not know which EHR entries belong to one patient. However, the cloud may know more about the pseudonyms of one patient than the common attackers outside. The patient or the doctor treating the patient may have to access all the patient's EHR entries in one query. They have to send all the pseudonyms of the patient to the cloud for retrieving all the corresponding EHR entries. If the cloud wants to, it can record the pseudonym set from a single querying source (e.g., network IP). Although in general, it is difficult to prevent the cloud from gaining the pseudonym set knowledge, some tricks can be used to let the cloud not know the exact pseudonym set. E.g., the patient or doctor can send some fake pseudonyms (fake pseudonyms mean the pseudonyms existing in the cloud but belonging to other patients. As we assume the pseudonyms in the cloud are public available resources, anyone can view these pseudonyms freely.) to the cloud with the actual pseudonyms; the patient or doctor can use anonymous proxies and other network technologies to avoid to be traced.

5.1.2 Dealing with Pseudonym Collision

In the above pseudonym scheme, the secret parameters (MSKs) for generating pseudonyms for patients are chosen by the patients themselves independently. It is possible that two patients choose a same parameter, although the probability is very low. This would certainly cause problems if it happened, as the pseudonyms generated by the two patients will be equal. A solution is to utilize PID_0 to avoid the occurrence. The system provider of the smart card can control that all patients will have different PID_0 s, or the PID_0 can be set by each patient to be one of the patient's unique identifiers (e.g., identity no., passport no., SSN, etc.). According to the algorithm of pseudonym generation, it is expected that different pseudonym sequences will be produced. Although this solution can prevent obvious collision of two pseudonym sequences of two patients, there is still very small probability that two arbitrary pseudonyms (from one patient or two patients) might collide since the generating scheme is not an injection. A solution is to count on the encryption or HMAC (Keyed Hashing for Message Authentication Code) on EHR data. That is, the collision of two pseudonyms is allowed by the cloud. But the EHR data must have been encrypted with a detection of an error or have a HMAC by different secret keys. The two EHR entries with same pseudonym are highly expected to have different encryption keys or different HMACs by using the pseudonym scheme designed in this paper, so the patients can recognize which one is the desired one upon receiving more than one answer from the cloud when they retrieve the EHR entries by providing a pseudonym to the cloud. The collided EHR entries which cannot be correctly decrypted or cannot get a correct HMAC will be dropped as undesired EHR entries by the patients. As for the possible repeated occurrence of pseudonyms, the cloud should not use the pseudonym section alone as the main index key for the database table.

Another problem caused by the possible collided pseudonyms is that, as the cloud cannot tell which patient is the exact owner of two EHR entries with one collided pseudonym from two different patients, a patient could update the other patient's EHR entry which does not belong to him/her. One solution is a patient only creates updates in new EHR entries with reference to the updated entries. However, in many cases, the cloud could do some further check to confirm whether the patient is making a valid updating on the EHR entry with a collided PID or not. For example, in the case where a patient wants to update the prescription's signature section, the update should include a valid signature on the prescription from a pharmacist, so the cloud can check the pharmacist's signature to prevent arbitrary updating by the patient. While in other cases, where the updates are made by the patient, (e.g., to set custom access control) the updates should be encrypted or have some integrity check mechanism. The malicious updates can be abandoned by the real owner of the EHR entry, even if the cloud did not carry out the further check or wrongly accepted some malicious updates.

5.1.3 Trust Mode of the Cloud

The cloud in this paper is not necessary to be trusted, but it is expected to be honest. This is usually the case in practice. The physical storages of public clouds are distributed and the data stored in them may be vulnerable for abuse by the cloud itself or easily be obtained by a skilled attacker. In this paper, we do not store in the cloud more information (such as patient's registration information) than protected data or publicly available data (e.g., root certificates of trusted authority for validating doctors). But we require that the cloud acts honestly, i.e., the healthcare application software running deployed by the "system provider" on the cloud is honest. For example, as the cloud does not intend to tamper the EHR data, it will also follow the protocol designed for authenticating the ownership of the EHR entries before allowing an update on them. We also require the cloud to understand some structure information of the EHR entries and to process just the desired sections (e.g. the section of a pharmacist's signature on the prescription) of the EHR entries in an honest manner. If the protocols are properly designed, the cloud cannot benefit from dishonesty except disturbing the normal procedure of the activities in eHealth system.

5.2 Computation Capability of the Smart Card

The pseudonym scheme in this paper might bring too much computation to the smart cards of the patients if all the computations need to be done in the smart cards. As shown in the section 4, one single pseudonym generation or reproduction needs three hash computations and two modular exponentiation computations, where hash computation is usually fast but the modular exponentiation computation is more time consuming. Noticing that the last two hash computations and the first modular exponentiation computation do not involve the secret *x*, they can be done by outside connected devices (e.g., the doctor's or the patient's computer). In a typical implementation of the modular exponentiation computation (Schneier 1996), about two thirds of the modular multiplication computations in the second modular multiplication computation will not involve the secret power *x*, so these modular multiplication computations can also be moved to outside devices. Thus, the computations needed to be done by the smart cards are only one hash computation and part of one modular multiplication computation. It greatly decreases the computation burden of the smart cards with a bit increase of the serializable IO manipulations.

5.3 Recovery upon Smart Card Loss

The patients' secrets MSKs and PINs should have a secure backup in the custody of the patients in case the smart cards are lost, since these secrets are only known by the patients, nobody else can recover these secrets instead. Once a patient lost his smart card, he needs to get a new one from the system provider, and restore the backup secrets into the new card. Then he needs to re-register at the insurance company and get a new certificate with a new corresponding private key. The EHR data stored on the cloud can be reused by the new card afterwards. The order number *last* of PID_{last} can be recovered by the patient sending the reproduced pseudonyms from the first one to the last one upon which the cloud can return correct EHR data. However, the feasibility of having all patients keeping binary secrets depends on the practical condition. Some feasible solutions are available. E.g., patients can use some public service (e.g. network storage service) to store their secrets. The general doctors of the patients could also help keeping the secrets. The secrets stored at public storage or general doctors' computers could be protected by patients-only known passwords. Although the security of the passwords is limited, it can release the complete trustworthiness on the general doctors or public storage service providers a little.

In some particular case, if the PIN and the smart card are taken by some attacker other than the patient self, but the MSK stored in the smart card is not known to the attacker (this is the common case in practice), the attacker could access the patient's existing EHR entries. The patient who knows the MSK can cooperate with the cloud to change the EK and even MSK to disable the old EHR entries. If the MSK is also compromised by the attacker (this is very rare in practice), the attacker could certainly do what the patient can. The only possible way is that the patient cooperates with the cloud to change the MSK and EK to disable the old EHR entries in time (before the attacker could do that).

6 CONCLUSIONS

In this paper, we presented how to use patients' pseudonyms for indexing, encrypting, and verifying the ownership of the EHRs with patients' privacy properly protected in cloud-based eHealth systems. A new decentralized pseudonym scheme was proposed accordingly. The security of the scheme is provable as it is based on the discrete logarithm problem and the one way property of hash functions. Although the security of the scheme can be proved, it faces much vulnerability in healthcare activities in cloud-based eHealth systems, which are complex systems involving many entities and complicated application scenarios. In comparison with the existing pseudonym schemes, our proposed scheme is dedicated for cloud-based eHealth systems. We minimize the data that the patients need to store on the smart cards to generate a new PID or to reproduce previous PIDs. We also lower the dependency on the trustworthiness of the cloud, and move the trust base to patients themselves and some technical assumptions (e.g., the smart card with protected memory and authentication PIN). Some corresponding protocols are carefully designed to apply the pseudonym scheme into cloud-based eHealth systems. We discussed several common application scenarios, and pointed out some possible problems (e.g., PID collision, smart card loss, etc.), and attacks (e.g., malicious cloud, ISP, insurance companies, eavesdroppers, etc.), with corresponding countermeasures. We also provide an optional scheme for decreasing the computation burden in the smart cards. These would make the proposed pseudonym scheme more feasible to be implemented in practical eHealth systems.

As future work, the computational cost of the pseudonym scheme needs to be evaluated in prototype eHealth systems equipped with simulated cloud, smart cards and users. What is more, we will continue to investigate the proposed pseudonym scheme as to feasibility and compatibility with other technologies in practical eHealth systems.

ACKNOWLEDGEMENTS

We would like to thank Adrian Spalka who raised many challenges and comments from the viewpoint of practical eHealth systems to this paper. Also we appreciate Jan Lehnhardt for discussing with us about the details of the scheme. What is more, he commented on the early version of this paper and helped with much literal work.

REFERENCES

Alemán, J. L. F., Señor, I. C., Lozoya, P. Á. O., Toval, A., 2013. Security and privacy in electronic health records: A systematic literature review. *Journal of biomedical informatics*.

Alhaqbani, B., Fidge, C., 2008. *Privacy-preserving electronic health record linkage using pseudonym identifiers*. HealthCom 2008. 10th International Conference on e-health Network- ing, Applications and Services: IEEE. pp. 108-117.

Deng, M., Petkovic, M., Nalin, M., Baroni, I., 2011. *A Home Healthcare System in the Cloud--Addressing Security and Privacy Challenges*. 2011 International Conference on Cloud Computing: IEEE. pp. 549-556.

Garets, D., Davis, M., 2006. Electronic medical records vs. electronic health records: yes, there is a difference. *Policy white paper*. Chicago, HIMSS Analytics.

Li, Z.-R., Chang, E.-C., Huang, K.-H., Lai, F., 2011. *A secure electronic medical record sharing mechanism in the cloud computing platform.*, 15th International Symposium on Consumer Electronics (ISCE): IEEE. pp. 98-103.

Lim, C. H., Lee, P. J., 1997. A key recovery attack on discrete log-based schemes using a prime order subgroup. In *Advances in Cryptology—CRYPTO'97*, pp. 249-263: Springer.

Löhr, H., Sadeghi, A.-R., Winandy, M., 2010. *Securing the e-health cloud*. Proceedings of the 1st ACM International Health Informatics Symposium: ACM. pp. 220-229.

Lysyanskaya, A., Rivest, R. L., Sahai, A., Wolf, S., 2000. Pseudonym systems. *Selected Areas in Cryptography*: pp. 184-199.

McCurley, K. S., 1990. *The discrete logarithm problem*. Proc. of Symp. in Applied Math. pp. 49-74.

Mell, P., Grance, T., 2011. The NIST definition of cloud computing (draft). *NIST special publication*, 800 (145): 7.

Microsoft., 2007. HealthVault. www.healthvault.com.

Pfitzmann, A., Köhntopp, M., 2001. *Anonymity, unobservability, and pseudonymity—a proposal for terminology*. Designing privacy enhancing technologies: Springer. pp. 1-9.

Rui, Z., Ling, L., 2010, 5-10 July 2010. *Security Models and Requirements for Healthcare Application Clouds*. IEEE 3rd International Conference on Cloud Computing: IEEE. pp. 268-275.

Ruland, C. M., Brynhi, H., Andersen, R., Bryhni, T., 2008. Developing a shared electronic health record for patients and clinicians. *Studies in health technology and informatics*, 136: 57-62.

Schneier, B., 1996. *Applied cryptography. Protocols, Algorithms, and Source Code in C/Bruce Schneier*: John Wiley, Sons, Inc.

Stingl, C., Slamanig, D., 2008. Privacy-enhancing methods for e-health applications: how to prevent statistical analyses and attacks. *International Journal of Business Intelligence and Data Mining*, 3 (3): 236-254.

Tang, P. C., Ash, J. S., Bates, D. W., Overhage, J. M., Sands, D. Z., 2006. Personal health records: definitions, benefits, and strategies for overcoming barriers to adoption. *J Am Med Inform Assoc*, 13 (2): 121-6.