Transitive Pseudonyms Mediated EHRs Sharing for Very Important Patients

Huafei Zhu(✉) and Ng Wee Keong

School of Computer Science and Engineering, Nanyang Technological University, Singapore, Singapore
huafeizhu@gmail.com, awkng@ntu.edu.sg

Abstract. Electronic health record (EHR) greatly enhances the convenience of cross-domain sharing and has been proven effectively to improve the quality of healthcare. On the other hand, the sharing of sensitive medical data is facing critical security and privacy issues, which become an obstacle that prevents EHR being widely adopted. In this paper, we address several challenges in very important patients' (VIPs) data privacy, including how to protect a VIP's identity by using pseudonym, how to enable a doctor to update an encrypted EHR with the VIP's absence, how to help a doctor link up and decrypt historical EHRs of a patient for secondary use under a secure environment, and so on. Then we propose a framework for secure EHR data management. In our framework, we use a transitive pseudonym generation technique to allow a patient to vary his/her identity in each hospital visit. We separate metadata from detailed EHR data in storage, so that the security of EHR data is guaranteed by the security of both the central server and local servers in all involved hospitals. Furthermore, in our framework, a hospital can encrypt and upload a patient's EHR when he/she is absent; a patient can help to download and decrypt his/her previous EHRs from the central server; and a doctor can decrypt a patient's historical EHRs for secondary use under the help and audit by several proxies.

Keywords: Electronic health record · Pseudonym · Semantic security · Transitive pseudonym

1 Introduction

Medical tourism is in great demanding [1–7], and since the medical tourists are often very important persons, healthcare service providers may be more likely to care for VIPs such as celebrities, super star athletes, and political leaders and many efforts have been proposed to handle the VIP medical records which are not shared across the public sectors currently in many countries and areas. The designation of VIPs in healthcare usually refers to the patients with great concern for their privacy and confidentiality due to the very high public profiles. To facilitate inter-professional collaboration and to enable disease management

J. Miller et al. (Eds.): ICWS 2019, LNCS 11512, pp. 80–94, 2019.
https://doi.org/10.1007/978-3-030-23499-7_6

and optimal technology support, one may handle VIPs' medical records within a public healthcare sector so that the outsourced EHRs can be accessed anytime, anywhere and anyhow for legitimate users. As important as it is to protect VIPs from bodily harm during the visit, it is equally important to protect them from attacks on the confidentiality via unauthorized access to the electronic medical records and to prevent the identifications of VIPs on various systems being traced [8–13].

1.1 The Motivation Problem

Considering a scenario where a VIP Alice visits a clinic for a medical treatment. She does not want to show the real identity to the clinic, as she is afraid of the exposure of her private medical condition caused by system attacks or reveal from the administrative staff or doctor in the clinic. She does not want to use the same pseudonym for every visit either, because the frequency she visits the clinic, the statistics in medical reports, and the linkage among different treatments and tests (e.g., blood test, X-ray, etc.) may help someone, who has certain background knowledge of Alice, guess her identity from those data. She would like to use a different pseudonym every time she visits the clinic. When Alice registers at the clinic with a new pseudonym, the administrative staff must verify that this pseudonym corresponds to a legal resident. Later when Alice sees a doctor Bob, Bob will pull all Alice's previous EHRs, under her help, from the central server to assist diagnosing, though those EHRs are indexed by different IDs (e.g., Alice's different pseudonyms). Two days later, when a new blood test report is out, Bob needs to update Alice's EHR in the central server, with Alice's absence. One month later, Bob finds Alice's case is special and worth of further research. Then Bob downloads and decrypts all Alice's EHRs generated before her last consultation with him, under the help of some authorities. To summarize the above scenario, when the VIP Alice registers in a medical center, she should not use the real identity, nor the same pseudonym as that was used in previous visits to this medical center. In other words, the local database in a medical center should be private enough to avoid data linkage in case it is compromised or some insider (e.g., administrative person) intentionally or unintentionally reveal the patient data. Furthermore, it can minimize the threat if the EHRs of a person are indexed by different IDs in the central server that are not linkable by adversaries. On the other hand, when a doctor tries to pull historical EHRs of a patient from the central server, he/she must be able to identify which records belonging to this patient and decrypt them for reading. This process sometimes needs to be achieved with the patient's absence, i.e., the patient cannot help to identify and decrypt his/her EHRs.

To achieve this goal, we need to solve the challenges:

- A way to generate pseudonym so that the real identity of a patient cannot be linked with the pseudonym;
- A way to generate multiple non-repeated pseudonyms from a patient's identity, and to verify these pseudonyms belong to the same identity;

– A way to prove to the administrative person during registration that the pseudonym to be registered corresponds to a legal resident;
– A way to search and decrypt all encrypted EHRs of a certain patient, which are indexed by different IDs, in a central server for an authorized action like EHR update or secondary use, when the patient is absent.

1.2 The Related Work

VIPs's EHR data stored in a server should be carefully encoded and protected. The existing cryptographic techniques guarantees the security of encrypted EHR data. However, the process of decryption should consider the absence of patients. In the motivation problem scenario, many medical test results and diagnoses are made when a patient leaves a medical center, so his/her EHR should be updated by the doctor. For the secondary use of EHRs (mainly for research purposes, as agreed in HIPAA) requires a doctor to decrypt certain EHR data. Such research issues have not been well addressed. To protect personal information of VIPs, the de-identification should be introduced. The state-of-the-art pseudonyms can be categorized into two types: irreversible and reversible pseudonyms. Irreversible pseudonyms are pseudonyms that cannot be reversed back to the original data owners' identities. Reversible pseudonyms are pseudonyms that can be reversed back to the back to the original data owners' identities. If one wants to get irreversible pseudonyms, then one-way functions such as cryptographically secure hash functions that modelled as random functions are applied. [14,15] and [16] use hash function to generate pseudonyms and assign a unique pseudonym to each patient while the identity information from every data packet is masked and chopped off in [17]. The l-diversity solution [18] and k-anonymity solution in [19] are fallen into the category of irreversible pseudonyms. If one wants to get reversible pseudonyms, then trap-door permutations such as cryptographically secure symmetric key encryptions are applied. In [20], a trusted third party based solution for generating reversible pseudonyms is presented and analyzed. And in [21], an interesting hardware and software concept is presented which allows the reversible and irreversible encryption of sensitive sample data without the need of electronic connectivity. Although, both irreversible and reversible pseudonyms provide protections of patients' data so that any information held by healthcare providers cannot be linked to an individual, in practice, there are times when for legitimate reasons multiple de-identified records of the same patient may need to be linked, for example, when we need to study the history of a patient medical condition. To guarantee an individual transaction associated with a different pseudonym, we also require that the generated pseudonyms can be efficient re-randomized.

1.3 This Work

In this paper, we propose a novel framework that is secure for VIP EHR data protection and convenient for EHR data sharing among authorized parties. In our framework, there is a Information Center in each hospital that stores the

metadata of the EHR generated by the doctors in this hospital, and takes the role to communicate with the central server and the Information Centers in other hospitals for EHR data searching, encryption, decryption and sharing. A doctor's role is as simple as submitting diagnosis to and request for historical medical reports from the Information Center, without caring about EHR searching, encryption, decryption and sharing. A patient can use a different pseudonym in each hospital visit, and controls the public/secret key for the encryption and decryption of his/her EHR.

Organization: the rest of this paper is organized as follows. Section 2 presents technical details on pseudonym generation and verification. Section 3 describes how encryption and decryption are done by the user-controlled public/secrect key. The repository designs for the metadata in each hospital's Information Center and the encrypted EHR data in the central server, as well as how cross-domain queries are performed are discussed in Sect. 4. Finally, we conclude this paper in Sect. 5.

2 Syntax and Security Definition of Pseudonym Generators

Definition 1. *A pseudonym generator $\mathcal{N}$: $D \times A \rightarrow R$ is a probability polynomial time (PPT) algorithm such that on input $id_U \in D$ and $aux \in A$, $\mathcal{N}$ outputs a pseudonym $nid_U \in R$, i.e., $nid_U = \mathcal{N}(id_U, aux)$, where D is a domain of user identities, A is an domain of auxiliary strings and R is a domain of pseudonyms.*

$\mathcal{N}$ is irreversible if for every probabilistic polynomial time adversary $\mathcal{A}$, there exists a negligible function $neg(1^\kappa)$ such that $Pr[id_U \leftarrow \mathcal{A}(nid_U, \perp) | nid_U = \mathcal{N}(id_U, aux)] \leq \text{neg}(1^\kappa)$, where κ is a security parameter and $\perp$ stands for an empty auxiliary string.

$\mathcal{N}$ is reversible if on input a valid nid_U there exists an efficient identity extractor $\mathcal{X}$ such that id_U can be extracted with the overwhelming probability, i.e., $Pr[id_U \leftarrow \mathcal{X}(nid_U, aux)] \geq 1 - \text{neg}(1^\kappa)$, where κ is a security parameter.

To define the security of the pseudonym generator, we consider the following experiment running between $\mathcal{N}$ and an adversary $\mathcal{A}$:

- $\mathcal{N}$ provides $\mathcal{A}$ black-box accesses to the sampling algorithm and the evaluation algorithm for polynomial many pairs $(id_{U_1}, nid_{U_1}), \ldots, (id_{U_l}, nid_{U_l})$, where $id_{U_i} \in D$ is adaptively selected and the adversary $\mathcal{A}$ obtains the corresponding pseudonyms $\{nid_{U_i}\}_{i=1}^{l}$;
- $\mathcal{A}$ outputs two identities id_{x_b} and $id_{x_{\bar{b}}}$. $\mathcal{N}$ randomly selects a bit $b \in \{0, 1\}$ and gives nid_{x_b} to $\mathcal{A}$. Adversary wins the game if it outputs a bit b' such that $b' = b$.

Definition 2 *(Semantic security of pseudonym generators): Let $Adv(\mathcal{A}) = |Pr(b' = b)| - 1/2$. A trapdoor pseudonym generator is semantically secure against chosen-identity attack if $Adv(\mathcal{A})$ is at most a negligible amount.*

We provide an efficient construction of pseudonym generator in the common reference string model, where no trusted third party assumption is made. Our pseudonym generator is based on the ElGamal encryption scheme. Let p be a large prime number, i.e., $p = 2q+1$, p, q are larger prime numbers. Let $G =< g >$ be a cyclic group of order q and H: $\{0,1\}^* \rightarrow Z_p^*$ be a cryptographic hash function. The idea behind our construction is that we allow a pseudonym generator and its insurance company or any other external auditor to collaboratively generate a verifiable common reference string h such that the discrete logarithm $\log_g(h)$ is unknown to all participants and then we will define a pseudonym of user U as a ciphertext $(u = g^r, v = \mathsf{H}(id_U)^2 \times h^r)$ of user's id together with a proof that id_U is encrypted by a Diffie-Hellman quadruple (g, h, g^r, h^r) in zero-knowledge (*notice that* id_U *should be encoded in the form of* $(\mathrm{H}(\mathrm{id_U}))^2$ *mod p rather* $\mathrm{H}(\mathrm{id_U})$ *mod p since the value* $\mathrm{H}(\mathrm{id_U})$ *mod p may not be an element in* G).

Definition 3. *Two ensembles* $X = \{X_n\}_{n\in N}$ *and* $Y = \{Y_n\}_{n\in N}$ *is called statistically close if the statistical difference is negligible, where the statistical difference (also known as variation distance) of* X *and* Y *is defined as* $\Delta(n) = 1/2 \sum_\alpha |\Pr(X_n = \alpha) - \Pr(Y_n = \alpha)|$.

2.1 Root Pseudonym Generators

Based on the generated common reference string $h \in G$, a user (say, Alice) first provides her National Registration Identity Card (NRIC) to the certificate authority (CA) for verifying that she is a genuine holder of this NRIC (*we identify user U's NRIC with* id_U *throughout the paper*). If the check is succeed (the validity of a user's NRIC can be verified via a physical contact or a secure channel that has been established between user and the pseudonym generation center), Alice generates a cipher-text $c = (u, v)$, where $u = g^r \bmod p$ and $v = h^r \times (\mathrm{H(NRIC)})^2 \bmod p$ of NRIC using ElGamal encryption scheme, and then proves in zero-knowledge that she knows the plain-text $(\mathrm{H(NRIC)})^2$ of the corresponding cipher-text c. If the proof is valid, then CA issues a certified pseudonym (g, h, g^r, h^r) to Alice. The procedure for generating a pseudonym nid_U of a user U (say, Alice) is described as follows:

- Alice first demonstrates her ownership of the presented NRIC that will be used for applying for her initial pseudonym; If the check is valid, Alice then randomly computes a Diffie-Hellman quadruple (g, h, u, v), where $u = g^r \bmod p$ and $v = h^r \times (\mathrm{H(NRIC)})^2 \bmod p$.
  Let $w \leftarrow \frac{v}{(\mathrm{H(NRIC)})^2}$ and let π be the following zero-knowledge proof of knowledge that the generated (g, h, u, w) is a Diffie-Hellman quadruple
  - Alice randomly selects $u' = g^{r'}$, $w' = h^{r'}$ and sends (u', w') to the pseudonym generator center $\mathcal{N}$;
  - $\mathcal{N}$ randomly selects a challenge string $e \in Z_q$ uniformly at random and sends it back to the prover Alice;
  - Upon receiving e, Alice computes the response $r'' = r' + er \bmod q$, and sends r'' to $\mathcal{N}$.

Let π be a transcript of the above interactive proof $< (u', w'), e, r'' >$

– Upon receiving (u, v) and π, the pseudonym issue checks the valid of π using the auxiliary information NRIC by the following check: $g^{r''} \stackrel{?}{=} u'u^e \bmod p$ and $h^{r''} \stackrel{?}{=} w'w^e \bmod p$; else it outputs 0 and terminates the protocol; If the proof is valid, then a certificate $cert_U$ on the pseudonym $\mathsf{nid}_U \leftarrow (u, v)$ will be issued.

We call $(nid_U, cert_U)$ be a root pseudonym of user U. We stress that the zero-knowledge proof should be performed and verified in a secure and authenticated channel since any leakage of zero-knowledge proof results in the user can be traced back efficiently. It is clear that the proof π is zero-knowledge and the proposed pseudonym generator is semantically secure assuming the decisional Diffie-Hellman problem defined over prime filed Z_p^* is hard. As a result, the proposed pseudonym is zero-knowledge and it is semantically secure assuming the decisional Diffie-Hellman problem defined over prime filed Z_p^* is hard.

2.2 Transitive Pseudonyms

In this section two methods for attesting pseudonyms are introduced and formalized: directly and indirectly attested pseudonyms. Informally, a directly attested pseudonym is a new pseudonym generated at session α which is attested by the previous pseudonym generated at session $\alpha-1$. An indirectly attested pseudonym is a new pseudonym generated at session α which is attested by a previously generated pseudonym generated at session β ($\beta < \alpha - 1$).

Direct Attestation. As usual in the real world, a user should register in the front desk before he/she is treated by a healthcare provider. during this initial registration, a user should demonstrate the healthcare provider that he/she is a valid user registered in a pseudonym distribution authority so that in case that a dispute occurs, the id of this registered user can be revealed (under what condition, an id of the registered user should revealed by law is out of the scope of this paper).

In our setting, a user will simply show that she/he is a genuine holder of $<\mathsf{nid}_U, \mathsf{cert}_U>$. The user then refreshes the initial pseudonym for the current registration. Let $(g, h, g^{r'}, h^{r'}(\mathrm{H(NRIC)})^2 \bmod p)$ be a pair of refreshed pseudonym. The user proves in zero-knowledge the fact that $(g, h, g^r, h^r(\mathrm{H(NRIC)})^2 \bmod p)$ and $(g, h, g^{r'}, h^{r'}(\mathrm{H(NRIC)})^2 \bmod p)$ are ciphertexts of the same scratched user id to the healthcare registration desk. Let $\overline{u} = \frac{u'}{u}$ and $\overline{v} = \frac{v'}{v}$. Notice that the prover (Alice) knows r and r' and hence she knows the difference $\overline{r} = r - r'$ such that $\overline{u} = g^{\overline{r}}$ and $\overline{v} = h^{\overline{r}}$. Since $(g, h, g^r, h^r(\mathrm{H(NRIC)})^2 \bmod p)$ is a certified pseudonym, it follows that we need to prove the following two things: (1) A proof of knowledge r such that $u = g^r$ in $<\mathsf{nid}_U, \mathsf{cert}_U>$ and (2) a proof of knowledge such that $(g, h, \overline{u}, \overline{v})$ is a Diffie-Hellman quadruple. The details of performs are processed below

Protocol 1: A proof of knowledge r such that $u = g^r$ in $<\mathsf{nid}_U, \mathsf{cert}_U>$

- Alice chooses a random string $s \in [1, q-1]$ and computes $\widehat{u} = g^s$, then sends $\widehat{u}$ to the healthcare provider H;
- H chooses a random string $f \in [1, q-1]$ uniformly at random and sends f to Alice;
- Alice computes $t = s + rf \bmod q$ then sends t to H
- H checks the equation: $g^t = \widehat{u}u^f \bmod p$. If the condition is satisfied then accept otherwise reject.

Protocol 2: A proof of knowledge such that $(g, h, \overline{u}, \overline{v})$ is a Diffie-Hellman quadruple

- Alice randomly selects $\overline{u}' = g^{\overline{r}'}$, $\overline{v}' = h^{\overline{r}'}$ and sends $(\overline{u}', \overline{v}')$ to the pseudonym generator center $\mathcal{N}$;
- $\mathcal{N}$ randomly selects a challenge string $e \in Z_q$ uniformly at random and sends it back to the prover Alice;
- Upon receiving e, Alice computes the response $\overline{r}'' = \overline{r}' + e\overline{r} \bmod q$, and sends $\overline{r}''$ to $\mathcal{N}$.
- H checks the equation: $g^{\overline{r}''} = \overline{u}'\ \overline{u}^e \bmod p$ and $h^{\overline{r}''} = \overline{v}'\overline{v}^e \bmod p$. If the condition is satisfied then accept otherwise reject.

Let π be the concatenation of transcripts of Protocol 1 and Protocol 2. We show that the successful proof π guarantees (g, h, u', v') and (g, h, u, v) are encryptions of the same NRIC.

Theorem 1. *The newly generated pseudonym (g, h, u', v') and the previously generated pseudonym (g, h, u, v) are encryptions of the same NRIC.*

Proof. One can verify that both Protocol 1 and Protocol 2 are complete, sound and zero-knowledge. By Protocol 1, we know that Alice is the genuine holder $<\mathsf{nid}_U, \mathsf{cert}_U>$ since Alice proves her knowledge r such that $u = g^r$, where (g, h) are common reference strings such that the knowledge of $\log_g h$ is unknown to all participants.

By Protocol 2, we know that $(g, h, \overline{u}, \overline{v})$ is a Diffie-Hellman quadruple. This means that there exists a value $\delta \in Z_q$ such that $\overline{u} = g^\delta$ and $\overline{v} = h^\delta$. Recall that $\overline{u} = \frac{u'}{u}$ and $\overline{v} = \frac{v'}{v}$, it follows that $u' = ug^\delta = g^{r+\delta}$ and $v' = vh^\delta = h^{r+\delta}(\mathrm{H}(\mathrm{NRIC}))^2$ (which is guaranteed by $<\mathsf{nid}_U, \mathsf{cert}_U>$). As a result, (u', v') is a random refreshment of (u, v) such that both ciphertetxts are encryptions of the same plaintext NRIC.

Let π be a zero-knowledge proof of the statement that $(g,\ h,\ g^r,\ h^r(\mathrm{H}(\mathrm{NRIC}))^2 \bmod p)$ and $(g,\ h,\ g^{r'},\ h^{r'}(\mathrm{H}(\mathrm{NRIC}))^2 \bmod p)$ are ciphertexts of the same scratched user id. Form Lemma 2, we know that the newly generated pseudonym $(g,\ h,\ g^r,\ h^r(\mathrm{H}(\mathrm{NRIC}))^2 \bmod p)$ binds the user's NRIC. Therefore, if the proof is accepted, the healthcare provider will issue a certificate $\mathrm{cert}_U^{(H)}$ to this newly generated pseudonym denoted by $\mathrm{nid}_U^{(H)}$. The internal database d_H

of the healthcare provider H stores the following table privately. Notice that we store π in the internal repository since it is a proof that $\mathrm{nid}_U{}^{(H)}$ is valid and it is a witness for issuing a certificate $\mathrm{cert}_U{}^{(H)}$ and future auditing

$(\mathrm{nid}_U, \mathrm{cert}_U)$	$(\mathrm{nid}_U{}^{(H)}, \mathrm{cert}_U{}^{(H)})$	π

Indirect Attestation. The following are illustrative examples for the introduction of the indirect attestation:

- For a treatment in the hospital registered with $\mathrm{nid}_U^{(H_i)}$, there could be many related diagnoses such as blood test, X-ray check etc. If these diagnoses are outsourced using the same $\mathrm{nid}_U^{(H_i)}$, then these results can be linked trivially. To solve the problem, instead of re-generation a new pseudonym for patient Alice, we would like to re-randomize $\mathrm{nid}_U^{(H_i)}$ to get a new $\mathrm{nid}_U^{(H_{i,j})}$ for registered at the department $H_{i,j}$ in the hospital H_i and then proves to $H_{i,j}$ her knowledge that both $\mathrm{nid}_U^{(H_i)}$ and $\mathrm{nid}_U^{(H_{i,j})}$ are ciphertexts that bind the same NRIC.
- For a patient Alice who registered initially at hospital H_i, is now to register at hospital H_j. In this case, the user may chooses $(\mathrm{nid}_U, \mathrm{cert}_U)$ or $(\mathrm{nid}_U{}^{(H_i)}, \mathrm{cert}_U{}^{(H_i)})$ as a witness to attest a newly generated pseudonym $(\mathrm{nid}_U{}^{(H_j)}, \mathrm{cert}_U{}^{(H_j)})$ at H_j. If $(\mathrm{nid}_U, \mathrm{cert}_U)$ is used to attest $(\mathrm{nid}_U{}^{(H_j)}, \mathrm{cert}_U{}^{(H_j)})$, there exists an obvious linkage between $(\mathrm{nid}_U{}^{(H_i)}, \mathrm{cert}_U{}^{(H_i)})$ and $(\mathrm{nid}_U{}^{(H_j)}, \mathrm{cert}_U{}^{(H_j)})$ since both are attested by $(\mathrm{nid}_U, \mathrm{cert}_U)$. To resist the linkability attack, we would like to use $\mathrm{nid}_U^{(H_i)}$ to attest a newly generated pseudonym $\mathrm{nid}_U^{(H_i)}$. Such a procedure is called indirect attestation.

Based on the above discussions, we can now formalize the structure of indirect (recommended) attestations. Suppose that $(\mathrm{nid}_U, \mathrm{cert}_U)$ is used to attest $(\mathrm{nid}_U{}^{(H_1)}, \mathrm{cert}_U{}^{(H_1)})$, and $(\mathrm{nid}_U{}^{(H_1)}, \mathrm{cert}_U{}^{(H_1)})$ is used to attest $(\mathrm{nid}_U{}^{(H_2)}, \mathrm{cert}_U{}^{(H_2)})$ and so on. We can now construct an attestation path: $(\mathrm{nid}_U, \mathrm{cert}_U) \rightarrow (\mathrm{nid}_U{}^{(H_1)}, \mathrm{cert}_U{}^{(H_1)}) \rightarrow (\mathrm{nid}_U{}^{(H_2)}, \mathrm{cert}_U{}^{(H_2)}) \rightarrow \cdots \rightarrow (\mathrm{nid}_U{}^{(H_k)}, \mathrm{cert}_U{}^{(H_k)})$. Since each zero-knowledge proof binds the original NRIC, it follows that there is an indirect linkage between the root $(\mathrm{nid}_U, \mathrm{cert}_U)$ the destination node $(\mathrm{nid}_U{}^{(H_k)}, \mathrm{cert}_U{}^{(H_k)})$. This means that the concept of indirect attestation can be formalized in the notion of transitive graphs.

Definition 4. *A graph $G = (V, E)$ has a finite set V of vertices and a finite set $E \subseteq V \times V$ of edges. The transitive closure $G^* = (V^*, E^*)$ of a graph $G = (V, E)$ is defined to have $V^* = V$ and to have an edge (u, v) in E^* if and only if there is a path from u to v in G.*

In terms of the notion of transitive graph, a direct attestation is an edge between two nodes in a graph G while indirect attestation is an edge defined in its corresponding transitive closure graph G^*. As a result, the notion of indirect attestation can be viewed as a natural extension of the notion of the direct

attestation (if the number of intermediate nodes in a path is zero). We will further demonstrate that the introduced indirectly attested pseudonym is a useful tool for the cross-domain query of outsourced EHRs.

3 User Controlled Encryptions

In this section, a user controlled encryption scheme is proposed and analyzed. The idea behind our construction is that when a user U initially registers to a healthcare service provider H, the generated plain electronic health record is first processed in a private sector within H. This is because the main function of a doctor is to examine a patient's health and to find causes and solutions for the illness. As a consequence, a health record stored in the private database (managed by the healthcare provider) should only be accessible by the doctors and healthcare staffs who are involved in the treatment. An access to the user's EHR by a doctor or a healthcare staff is logged and audited. All healthcare professionals are also bound by law and professional ethics to keep user medical information strictly confidential during the period when the generated EHR are stored and processed in the private sector. To outsource the healthcare records, the plain EHR will be first encrypted by the user specified public key encryption scheme that is generated on the fly. The resulting ciphertexts are transferred to the public sectors. Below is a detailed construction of our protocol.

3.1 One-the-fly Public Key Generation

After the initial registration in H, a user Alice invokes a cryptographically strong pseudo-random generator G which takes the current state as an input and outputs $(s', k) \leftarrow G(s)$, where $s \in \{0,1\}^m$ in the current state and $s' \in \{0,1\}^m$ is the next state and $k \in \{0,1\}^m$ is the current output. We assume that Alice enciphers k as $K = g^k \bmod p$. K is then securely transferred via secure channel established between user and the healthcare provider (say, SSL via the web interface). The secret key sk used for enciphering the generated EHRs is encrypted by K.

3.2 Threshold Decryption

As mentioned earlier, in some cases a doctor needs to decrypt an EHR without the presence of the patient, e.g., the doctor needs to update the EHR or the doctor needs to use the EHR for further research (secondary use of EHR). In such cases, we need a practical decryption mechanism to obtain a plain EHR without a patient's assistance. In this paper, we will use the notion of threshold encryption.

In real world, enterprises often delegate the security verification of incoming people (to its premises) to companies which have specialized skill set in doing such job. Based on stated policies of an organization, these security companies verify various credentials of incoming people, before they are allowed to enter the

premises of the organization. We apply this approach in the context of threshold encryptions. Let $T_1, \ldots, T_m$ be m decryption proxies (servers) not necessary within the hospital (e.g., an insurance company, or a certificate issuing center can be designated decryption proxies). A user will process an l-out-of-m threshold public key encryption scheme for supporting, e.g., the secondary use of the encrypted EHRs:

- H randomly selects a polynomial $f(x) = f_0 + f_1 x + \cdots + f_{l-1} x^{l-1}$, where $f_0 = k$;
- Each processing center T_i is given a pair shares $(t_i, f(t_i))$ $(i = 1, \ldots, m)$, where t_i is an id of T_i;

We are able to provide an efficient solution to the problem stated in the motivation problem: Two days later, doctor Bob needs to update Alice's EHR that was encrypted and outsourced by the Information Center. One month later, Bob finds Alice's case is special and worth of further research. Then Bob downloads and decrypts all Alice's EHRs generated before her last consultation with him, under the help of some authorities. To serve such requests, i.e., the hospital H makes a decryption query of the ciphertext $(u, v) = (g^r, sk \times K^r)$, the Information Center will randomly select l decryption servers among the specified m proxies and send the corresponding u to the selected servers. Once the Information Center gets the m values $u^{f(t_i)}$ $(i = 1, \ldots, m)$, it can retrieve the plain EHR by the Lagrange interpolation formula.

If a public-key encryption secure against adaptively chosen ciphertext attack is used for encrypting a symmetric key sk that will be used to encipher the generated EHRs (say, the Cramer-Shoup's encryption scheme [22]), one needs to generate more randomness from k. An obvious solution is to invoke a new instance of the underlying pseudo-random generator G^* which takes as input k and runs recursively to output $(k_1, k_2, k_3, k_4, k_5, k_6) \in [1, q-1]^6$ such that $X = g^{k_1} h^{k_2} \bmod p$, $Y = g^{k_3} h^{k_4} \bmod p$ and $Z = g^{k_5} h^{k_6} \bmod p$. To encrypt a message $m = \mathrm{H}(\mathrm{NRIC})^2$, the Cramer-Shoup's encryption algorithm chooses $r \in [1, q-1]$ uniformly at random, then computes $u_1 = g^r \bmod p$, $u_2 = h^r \bmod p$, $v = mZ^r \bmod p$, $e = \mathsf{H}(u_1, u_2, e)$ and $w = X^r Y^{re} \bmod p$. (u_1, u_2, v, w) is called an encryption of the message m. The Canetti and Goldwasser's threshold public key cryptosystem secure against adaptive chosen ciphertext attack constructed from the Cramer and Shoup's encryption can be applied here. We refer to the reader [23] for more details.

4 Storage and Query

In our framework, the metadata of EHRs, i.e., the owner of each EHR and how each EHR links other EHR under the same owner are stored locally in the Information Center in each participating hospital. The encrypted EHR data are outsourced to the central server for sharing. This section describe how the EHR metadata and data storages are designed.

4.1 In-Hospital Repository

Although the encrypted EHR data are outsourced to the central server, the local database in the Information Center of each hospital should store some metadata about the EHRs generated in the hospital and their patient owners. The purpose of in-hospital data storage is to record the certified pseudonyms of the patients who accepted treatment in the hospital, and also to offer guidance to link each EHR to the previous EHRs of the same patient since this link is important for the secondary use of EHRs but should not be exposed in the central server.

Recall that an internal transcript generated during a patient's registration comprises of the following items: the previously generated and certified pseudonym $(\text{nid}_U{}^{(H_{i-1})}, \text{cert}_U{}^{(H_{i-1})})$, a newly generated and certified pseudonym $(\text{nid}_U{}^{(H_i)}, \text{cert}_U{}^{(H_i)})$, a proof π as well as a public-key K generated on the fly for EHR encryption. Thus the in-hospital database of a hospital H_i is designed as:

$$R_{H_i}(\textit{Pre-nym}, \textit{Pre-cert}, \textit{New-nym}, \textit{New-cert}, \textit{Proof}, PK)$$

Table 1 shows an example local database table for a hospital H_i.

Table 1. An example local database table for the hospital H_i

Pre-nym	Pre-cert	New-nym	New-cert	Proof	PK
nid_{U_1}	cert_{U_1}	$\text{nid}_{U_1}{}^{(H_i)}$	$\text{nid}_{U_1}{}^{(H_i)}$	$\pi_{U_1}^{(H_i)}$	K_{U_1}
$\text{nid}_{U_2}{}^{(H_j)}$	$\text{cert}_{U_2}{}^{(H_j)}$	$\text{nid}_{U_2}{}^{(H_i)}$	$\text{nid}_{U_2}{}^{(H_i)}$	$\pi_{U_2}^{(H_i)}$	K_{U_2}
...	...	...	...	...	...
$\text{nid}_{U_n}{}^{(H_k)}$	$\text{cert}_{U_n}{}^{(H_k)}$	$\text{nid}_{U_n}{}^{(H_i)}$	$\text{nid}_{U_n}{}^{(H_i)}$	$\pi_{U_n}^{(H_i)}$	K_{U_n}

We can see that each tuple contains the pseudonym of a patient that is certified by the CA or a previous visited hospital, and the new pseudonym certified by H_i. We need to emphasize that the proof π is also necessary to be stored. This is because each hospital will be aperiodically audited. The auditing will be done by checking the proof to show that the hospital is not cheating on each certified pseudonym. A tuple can be identified by either *New-nym* or *PK*. The different identifiers are used for searching and linking EHRs of a same patient under the circumstance that the patient is absent or present. The details will be covered in Sect. 6. Note that different pseudonyms will be used even if the same patient visits H_i multiple times.

The hospital will also temporarily store the EHR of a patient, before encrypting and uploading it to the central server. This period is rather short. Once the encrypted EHR is outsourced, the hospital should not maintain a copy. In the case that the database in a hospital is attacked, there will be no EHR exposed. Also, from the metadata, the adversary can only infer a one-level linkage of pseudonym, i.e., $\text{nid}_{U_2}{}^{(H_j)}$ and $\text{nid}_{U_2}{}^{(H_i)}$ are the same patient, but cannot reverse either his/her identity or medical condition.

4.2 Outsourced EHR Repository

The central server storage is designed as:

$$R\ (Nym,\ PK,\ EK,\ eEHR)$$

We assume that the secret key sk that will be used to encrypt the outsourced EHRs is created by the ElGamal encryption scheme. This is reasonable assumption since sk is one-time used for each K (similar with the notion of one-time encryption). When a personal health record d is created by a doctor in the hospital H, this data d will be encrypted by K using the ElGamal encryption scheme $c_K\ (:= (u_K, v_K)) = (g^r, sk \times K^r)$. The corresponding storage of the encrypted EHR is:

$nid_U^{(H)}$	K	c_K	$AES(sk,\ d)$

4.3 Patient-Aided Query

Now we consider the case that a patient Alice consults a doctor in a hospital, and needs to help the doctor find out all his/her previous EHRs in the central server, including the ones generated by other hospitals and the ones generated in this hospital before. In this case, Alice will invoke the pseudo-random number generator which takes a seed s as input and outputs (s', k). Recursively, she obtains $(k_1, \ldots, k_i)$ and hence obtains $K_1, \ldots, K_i$ accordingly as the public keys generated in previously visited hospitals. Then Alice searches the database in the central server using the generated index $K_1, \ldots, K_i$ to download all encrypted EHRs, and perform decryption.

4.4 Patient-Absent Query

As illustrated, sometimes a doctor needs to find out and decrypt a patient's EHRs when the patient is not around. If the doctor would like to update an EHR he/she produced before, he should submit the request to the Information Center, and the Information Center will help to download the EHR and decrypt it based on the threshold decryption. After updating the EHR, the Information Center will encrypt it and replace the one in the central server. Now we consider the secondary use of EHR. Assume a doctor would like to obtain all historical EHRs (that may be generated by different hospitals) of a patient for research purpose, he/she needs to do it collaboratively with the central server and the Information Centers of all involved hospitals. The procedure is shown in Algorithm.

Algorithm 1 Inter-hospital EHR searching

Input: the current eEHR d of the patient with pseudonym p in the hospital H, the database $DB_{H'}$ of an involved hospital H', the database DB_s of the central server
Output: a set S of all plain-text EHRs of the patient that were generated before d
1: let H decrypt d by threshold decryption and put the plain-text EHR into S
2: search DB_H for the previous pseudonym that was used to generate p by the patient, denoted by p'
3: **while** the authority that certified p' is not the CA **do**
4: let H' denote the hospital that certified p'
5: send request to H'
6: H' finds and decrypts the eEHR by p' from DB_s, and save the plain-text into S
7: H' search $DB_{H'}$ for the previous pseudonym used to generate p'
8: update p' as the newly found pseudonym
9: **end while**
10: return S as all the plain-text EHRs by the patient

Suppose a doctor D_i at hospital H_i and wants to query all the previous medical reports of a user U_2. Then the Information Center in H_i will search the local database, as shown in Table 1, and find that U_2 visited the hospital H_j before visiting H_i (from the second tuple in Table 1). Then the Information Center will send a request to the Information Center in H_j. The Information Center in H_j will go through a threshold decryption and send the EHR generated by H_j to H_i. Furthermore, H_j will recursively request another hospital for the previous EHR, until all EHRs of the patient are found.

5 Computation Complexity

All computations are measured with the multiplications mod$p$. Since a computation g^x mod$p$ on average is roughly $1.5\log_2[(p-1)/2]$ assuming that x is randomly distributed in $(p-1)/2$ using the standard square-and-multiplication method, where $p-1=2q$ and q is a large prime number. It follows that the computation complexity of Protocol 1 is $4.5\log_2(q)$ modular multiplications and Protocol 2 is $9\log_2(q)$ modular multiplications. Hence the transitive attestation is very efficient.

6 Conclusion

In this paper, a novel approach has been proposed and analyzed for securely handling the VIP EHRs to the public sector. Our method leverages the notion of trapdoor pseudonym generators. The transitive property of the proposed pseudonym generators benefits the healthcare professionals performing cross-domain queries efficiently. Our user controlled encryption protecting the outsourced healthcare records from attacks on the confidentiality and preventing the identifications of VIPs on various systems being traced.

References

1. Sadlier, C., Bergin, C., Merry, C.: Healthcare globalization and medical tourism. Clin. Infect. Dis. **58**(11), 1642–1643 (2014)
2. Ivanov, S., Webster, C., Mladenovic, A.: The microchipped tourist: implications for European tourism. Social Science Electronic Publishing (2014)
3. Beladi, H., Chao, C.C., Ee, M.S., Hollas, D.: Medical tourism and health worker migration in developing countries. Econ. Model. **46**, 391–396 (2015)
4. Fombelle, P.W., Sirianni, N.J., Goldstein, N.J., Cialdini, R.B.: Let them all eat cake: providing VIP services without the cost of exclusion for non-VIP customers. J. Bus. Res. **68**(9), 1987–1996 (2015)
5. Dang, H.S., Huang, Y.F., Wang, C.N.: Estimation of the market size of medical tourism industry using grey models - case study in South Korea. In: Estimation of the Market Size of Medical Tourism Industry Using Grey Models - Case Study in South Korea, pp. 46–50 (2016)
6. Arunotai, P.: An investigation of tourism information on destination management organization websites as the pull factor: a case study of health and wellness tourism information. In: 2017 11th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), pp. 1–8 (2017)
7. Zhao., H.: An investigation of tourism information on destination management organization websites as the pull factor: a case study of health and wellness tourism information. In: ICISS 2018 Proceedings of the 2018 International Conference on Information Science and System, pp. 102–106 (2018)
8. Yang, C.C., Leroy, G., Ananiadou, S.: Smart health and wellbeing. ACM Trans. Manag. Inf. Syst. **4**(4), 15:1–15:8 (2013). https://doi.org/10.1145/2555810.2555811
9. Yang, C.C.: Patient centered healthcare informatics. IEEE Intell. Inf. Bull. **15**(1), 1–5 (2014)
10. Yang, C.C., Veltri, P.: Intelligent healthcare informatics in big data era. Artif. Intell. Med. **65**(2), 75–77 (2015). https://doi.org/10.1016/j.artmed.2015.08.002
11. Spagnuelo, D., Lenzini, G.: Transparent medical data systems. J. Med. Syst. **41**(1), 8:1–8:12 (2017). https://doi.org/10.1007/s10916-016-0653-8
12. Daniels, M., Rose, J., Farkas, C.: Protecting patients' data: an efficient method for health data privacy. In: ARES 2018, Proceedings of the 13th International Conference on Availability, Reliability and Security (2018)
13. Alabdulhafith, M., Alqarni, A., Sampalli, S.: Customized communication between healthcare members during the medication administration stage. In: MobileHCI 2018 Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services (2018)
14. Riedl, B., Neubauer, T., Goluch, G., Boehm, O., Reinauer, G., Krumboeck, A.: A secure architecture for the pseudonymization of medical data. In: ARES, pp. 318–324 (2007)
15. Quantin, C., Jaquet-Chiffelle, D.O., Coatrieux, G., Benzenine, E., Fa, A.: Medical record search engines, using pseudonymised patient identity: an alternative to centralised medical records. Int. J. Med. Inf. **80**(2), 6–11 (2011)
16. Nugroho, H.A., Priyana, Y., Prihatmanto, A.S., Rhee, K.H.: Pseudonym-based privacy protection for steppy application. In: 2016 6th International Annual Engineering Seminar (InAES), pp. 138–143 (2016)
17. Sarkar, S., Chatterjee, S., Misra, S., Kudupudi, R.: Privacy-aware blind cloud framework for advanced healthcare. IEEE Commun. Lett. **21**(11), 2492–2495 (2017). https://doi.org/10.1109/LCOMM.2017.2739141

18. Shah, A., Abbas, H., Iqbal, W., Latif, R.: Enhancing E-healthcare privacy preservation framework through L-diversity. In: 2018 14th International Wireless Communications and Mobile Computing Conference (IWCMC) (2018)
19. Sweeney, L.: K-anonymity: a model for protecting privacy. Int. J. Uncertainty, Fuzziness Knowl.-Based Syst. **10**(5), 557–570 (2002)
20. Hillen, C.: The pseudonym broker privacy pattern in medical data collection. In: 2015 IEEE TrustCom/BigDataSE/ISPA, Helsinki, Finland, 20–22 August 2015, vol. 1, pp. 999–1005 (2015). https://doi.org/10.1109/Trustcom.2015.475
21. Ihmig, F.R., Wick, H., Hichri, K., Zimmermann, H.: RFID for anonymous biological samples and pseudonyms. In: 2011 IEEE International Conference on RFID-Technologies and Applications, RFID-TA 2011, Sitges, Spain, 15–16 September 2011, pp. 376–380 (2011). https://doi.org/10.1109/RFID-TA.2011.6068665
22. Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0055717
23. Canetti, R., Goldwasser, S.: An Efficient *threshold* public key cryptosystem secure against adaptive chosen ciphertext attack (extended abstract). In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 90–106. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_7