Pseudonymization for improving the Privacy in e-Health Applications

Bernhard Riedl, Veronika Grascher, Stefan Fenz, Thomas Neubauer
Secure Business Austria
Vienna, Austria
Email: riedl, grascher, fenz, neubauer@securityresearch.ac.at

Abstract— **Electronic health records (EHR) promise to improve communication between health care providers, thus leading to better quality of patients' treatment and reduced costs. As highly sensitive patient information provides a promising goal for attackers and is also demanded by insurance companies and employers, there is an increasing social and political pressure regarding the prevention of health data misuse. This paper presents a detailed description of the new system PIPE (Pseudonymization of Information for Privacy in e-Health) which differs from existing approaches in its ability to securely integrate primary and secondary usage of health data. Therefore, PIPE provides a solution to shortcomings of existing approaches. Our approach may be used as a basis for implementing secure EHR architectures or as an extension to existing systems.**

I. Introduction

In our today's health care system, the availability of sound information has tremendous impact on the decision regarding the patients' care and thus on the quality of treatment and patients' health. In this context, electronic health records (EHR) were introduced over the past several years as a method for improving communication between health care providers and access to data and documentation, leading to better clinical and service quality [1]. The EHR promises the reduction of adverse drug events (ADR) accounting for about $175 billion a year in the US [2] and for the very high number of more than 200.000 cases of death a year in the US [3] as it provides the physicians and their health care team [4] with decision support systems and guidelines for drug interactions. Further the EHR promises massive savings by digitizing diagnostic tests and images. A study by the nonprofit research organization Rand Corporation found out that adopting the EHR could result in more than $81 billion in annual savings in the US, if 90% of the health care providers used it [3].

However, the discussion of privacy is one of the fundamental issues in health care today and a trade-off between the patient's requirement for privacy as well as the society's needs for improving efficiency and reducing costs of the health care system. With informative and interconnected health-related data comes highly sensitive and personal information. As a result of the high sensitivity, there is an increasing social and political pressure to prevent the misuse of health data. On the one hand it is a fundamental right of every citizen to demand privacy and on the other hand the disclosure of medical data may cause serious problems for the patient. Insurance companies or employers could use this information to deny health coverage or employment. The disclosure of sensitive data, such as the history about substance abuse or HIV infection, could result in discrimination or harassment. In addition to social and political pressure, legal acts demand the protection of health data. Regarding the individual's privacy, historically the phrase "to be let alone", defined at the US Supreme Court in 1834, became famous. In 2006 the United States Department of Health & Human Service Health issued the Insurance Portability and Accountability Act (HIPAA) [5] which demands the protection of patients data that is shared from its original source of collection. Since 2005 the processing and movement of personal data is legally regulated by the EU with the Directive 95/46/EC [6]. A citizen's right of privacy is also recognized in the Article 8 [7] of the European Convention for the Protection of Human Rights and Fundamental Freedoms. Additionally, in the EU many domestic acts (e.g., the Austrian Data Protection Act [8]) dictate strict regulations on the processing of personal data.

Along with the improvement of interconnection, the increasing fear of data abuse as well as the adoption of laws lead to the development of a variety of techniques for protecting patients' identity and privacy. One possibility to assure patients' privacy is to encrypt the anamnesis data. As medical data tends to be very large (e.g., the image size of a x-ray is 6 MB, for a mammogram 24 MB [9] or for a computer tomography scan counts up to hundreds of MB [10]) and encryption is a highly time-consuming operation, the process to encrypt all data would not be manageable. As a consequence, several authors propose the usage of pseudonyms for privacy protection. The concept of pseudonymization (cf. [11], [12]) allows an association with a patient only under specified and controlled circumstances. Existing approaches (cf. [13]–[18]) have shortcomings like for example centralized patient-pseudonyms lists or the concealment of the applied algorithms. We will discuss these shortcomings in the Related Work section.

This publication presents a detailed description of PIPE (Pseudonymization of Information for Privacy in e-Health), a new system that differs from existing approaches in its ability to securely integrate primary and secondary usage (cf. [17]–[19]) of health data and thus provides a solution to security shortcomings of existing approaches. Moreover, it contains a new concept for data sharing, authorization and data recovery in case the user loses her access key. This approach can be

1530-1605/08 $25.00 © 2008 IEEE

used as a basis for implementing secure EHR architectures or as an extension to existing systems.

II. Related Work

Pseudonymization is a technique where identification data is transformed into a and afterwards replaced by a specifier which can not be associated with the identification data without knowing a certain secret [11], [12], [20]. As it is necessary for privacy reasons to avoid storing any personal information with the pseudonymized dataset, a pseudonymized database has to contain at least two tables, one where all the personal information is hold persistent, and another which keeps the pseudonyms and the pseudonymized data. The process of identifying and separating personal from related data is called depersonalization [21]. After depersonalization and subsequent pseudonymization, a direct association between certain persons and their data cannot be established. Algorithms for calculating the pseudonym may be based on encrypting or hashing techniques [22]. The latter demands to store a list where all pseudonyms are kept in order to assure reversibility [17], [18], [23], but relying on the use of a list is not secure, as an attacker, who gains access to this list, could establish an unauthorized relation between the identification and the medical data of a specific patient. Encryption provides a more secure alternative for building pseudonyms. For using encryption with a symmetric algorithm (e.g., AES [24]) a secret key, for the asymmetric alternative a key-pair (e.g., RSA [25], ECC [26]), is needed.

As demanded by Kerckhoffs' principle [27] only the keys have to be kept secret, whereas the applied algorithms are accessible. Hence, a major requirement for a secure system is that keys have to be shared with as few people as possible, preferable with nobody. Nowadays it is a common practice to store keys on smart cards [28], [29]. They are equipped with a small logic chip in order to conduct cryptographic operations without the need to process data on open systems like a standard client, for example a personal computer. This technique in combination with a certified card reader assures confidentiality and integrity (cf. [30] for a security taxonomy) of sensitive data during encryption and decryption. In other words, after authenticating against the smart card by entering a PIN, data is transfered to the card reader and afterwards processed on the card's cryptochip. If the PIN is only accessible to the cardholder, this technique can be considered secure [28], [29].

However, as smart cards may be lost, stolen, destroyed or compromised, it is a system's requirement to provide a fall-back mechanisms that allows recovering the key in order to re-establish access to the data which has been encrypted with the smart card. One approach is to keep all keys centralized within the system in a backup keystore which needs to be secured itself. Role-based access control (RBAC) models could be used for handling the authorization and authentication tasks of the backup keystore, but as role-based access control models can be by-passed or compromised [31], [32], a high level of security can only be established by encrypting the keystore itself [20]. Nevertheless, persons with administrative roles have to be granted access to the backup keystore for maintenance purposes [14], [20]. Therefore, this technique does not provide enough security for sensitive health data, because attacks could be performed by people working inside the system, e.g., by social engineering attacks [33]. In order to mitigate this vulnerability, threshold schemes [34] allowing to share keys between multiple administrators can be used. Another shortcoming of existing systems is the patients' dependence on a single pseudonym. If a patient only holds one pseudonym, an attacker who gains access to the database could use data mining [35] for identifying relations between medical data and the patient. Exemplarily, only a group of patients might have had a knee operation in a specific time slot. Moreover, perhaps only a few people of this group had been treated at a certain hospital and only one of them has seen her dentist a couple of days around the knee operation. Hence, as it is possible to conclude the identity of a person by combining single occurrences of her medical data, the usage of pseudonymization can only be considered secure if enough disjointed pseudonyms exist.

Several approaches for securing EHR architectures have been propsed. The system published by Thielscher et al. (cf. [14]) is based on decentralized keys stored on smart cards. Their approach consists of two databases, one for the patient's identification and one for the anamnesis data. The relation between a certain patient and her datasets can only be established by applying the necessary secret key on the smart card in order to generate the unique identifier. The system allows to authorize health care providers (HCP) to access specific anamnesis datasets. The major shortcoming of this systems is the dependence on a centralized patients-pseudonyms list, which provides a fall-back mechanism in case a patient looses her smart card because otherwise there would be no possibility to recover the mentioned identifier. Thielscher et al. circumvent this security flaw by operating the patients-pseudonyms list off-line. This organizational work-around promises a higher level of security until a social-engineering attack is conducted on a system's insider [33], [36] or an attacker gets physical access to the computer which holds this list. Pommerening et al. contributed two different approaches (cf. [17], [18]), which are both similar to the system of Thielscher et al. Their architecture, which is only applicable for the secondary use of medical data in research centers, is a combination of a hashing and an encryption technique. The encryption itself is based on a centralized secret key, which opens a vulnerability, because if an attacker knows this single key, she might gain access to all patients' related medical data. The approach of Peterson [13] is also based on a centralized table, which is used for reidentification purposes. This table which is, from a security point of view, comparable to the approaches of Pommerening or Thielscher relies on the same weak point, because a centralized list is attackable from in- or outside the system. In other words, it is a promising goal for any attacker. In 2001 another architecture was proposed by Schmidt et al. [37]. The underlying security of this system

is mainly based on encryption. Consequently the data is completely or partially encrypted, which in our opinion is too time-consuming to be applied for storing medical images.

Based on the mentioned security techniques and their shortcomings presented in this chapter, we define the following demands for a system that allows the secure pseudonymization of health care records:

1) Use depersonalization to divide all patient related information into two different tables or databases [21], one for the personal data and the second one for the anamnesis. This provides the basis for applying pseudonymization as confidentiality technique.
2) Replace the foreign key in the anamnesis database, which is related to specific persons, with a pseudonym [12], [22] to assure the patient's privacy.
3) To avoid data mining, every dataset combination of patient, HCP and anamnesis should be defined with a unique pseudonym [20]. For the same reason it is important to hide any relation between interacting persons.
4) Secure the keys to form pseudonyms and not the algorithm as demanded by Kerckhoff's principle [27].
5) Apply a threshold scheme, to share secrets like keys [34]. Moreover, conceal the association between the patients and their responsible administrators. This demand assures that no single person is able to unveil a certain person's identity.
6) Following the previous requirement, the number of administrators [34], which are assigned to hold a certain person's backup key and the number of administrators, which are necessary to act together to unveil the secret should be balanced.
7) Use role-based access control models only if it is not possible to control the access rights for a certain pseudonym by sharing encrypted secrets (e.g., keys or hidden relations) to access certain pseudonyms, because latter currently is the highest security level [20].
8) Provide the patient with the possibility to decide which datasets she wants to share by forming an unique pseudonym for the patient herself as well as for any patient-health care provider-anamnesis combination. In addition, hand over all rights to authorize or revoke persons, as far as possible as well as according to the legal situation [5]–[8], to assure that the patient is in full control of her data.

The following section introduces the generalized workflows of our prototype based on the demands stated in the enumeration above.

III. System Overview

PIPE consists of users $\mathcal{U}$, which are mapped to the roles patient, relative, health care provider and operator. The patient (A), as the owner of her data, is in full control of her datasets. Every patient may give one or more relatives (B) the right to access all of her anamnesis data. A health care provider or medical staff group (C) can be authorized to see or create a subset of the anamnesis data by the patient. The operators (O), as we named our administrative roles, share the secrets of the patients to provide a fall-back mechanism for lost, compromised or destroyed smart cards. Table 1 gives an overview of the keys and abbreviations used to describe our system. Note, that all private keys (where K stands for key) are identified as K^{-1}, for example the patient's inner private key will be named $\widehat{K}_A^{-1}$. All data is held persistent in the storage St, which represents the database as well as a secured keystore. In practice the logic L, and the storage St, which might be outsourced to a data processing center, have to form a trusted instance, because smart card management is handled there.

As shown in figure 1, PIPE is based on a hull-architecture [20], [38]. Every hull consists of one or more secrets (e.g., encrypted keys or hidden relations) which are only accessible with the unveiled secrets from the next outer hull. For instance the patient's inner private key $\widehat{K}_A^{-1}$ in the inner hull — or user permissions layer of the patient A — is encrypted with the outer public key K_A on her smart card, which represents the outer hull or authentication layer. A specific anamnesis dataset φ_i, which is associated with a list of j pseudonyms ψ_{i_j}, can only be accessed with the knowledge of the related secret, which has been encrypted with the inner symmetric key $\overline{K}_A$. As the inner symmetric key has been preliminary encrypted with the inner public key, this encryption operation has to be reversed to gain access to this key in plain-text. In other words, if a patient wants to access her data, she has to decrypt her inner private key $\widehat{K}_A^{-1}$, which is stored encrypted inside the system with the outer public key K_A of her smart card. Afterwards, she is able to decrypt the inner symmetric key $\overline{K}_A$ with her inner private key and can use the inner symmetric key, which is now available for her in plain-text, to access the encrypted secrets in the most inner hull – the concealed data hull – by decrypting them. Consequently, to get access to the data, every user has to 'peel the hulls'.

In our system PIPE, secrets can be shared between users for authorization purposes. First of all, a patient can provide a relative with her inner private key $\widehat{K}_A^{-1}$, which will then be encrypted with the relative's symmetric key $\overline{K}_B$. By doing this, the relative gets access to all data of the patient, until the inner private key is changed. Moreover, a health care provider can be authorized to access a subset of anamnesis datasets by sharing secrets, in our approach pseudonyms, of the concealed hull. A special case of a pseudonym, a so-called root pseudonym ψ_{i_0} exists for every dataset. This root pseudonym is only related with the patient and the anamnesis and no other user than the patient herself is able to delete this pseudonym. All other pseudonyms may be removed from the storage without authorized users permission. For example if two health care providers are related to see a specific anamnesis, three pseudonyms exist, whereas both pseudonyms which are shared between a patient and a health care provider may be deleted without the particular health care providers notification. If the patient decides to delete the anamnesis dataset, she is the only user, for whom it is possible to delete all pseudonyms. This assures that the patient is in full control

	Patient	*Relative*	*HCP*	*Operator*	*Logic*
abbreviation	A	B	C	O	L
unique identifier	A_{id}	B_{id}	C_{id}	O_{id}	
(outer public key, private key)	(K_A, K_A^{-1})	(K_B, K_B^{-1})	(K_C, K_C^{-1})	(K_O, K_O^{-1})	
(inner public key, private key)	$(\widehat{K}_A, \widehat{K}_A^{-1})$	$(\widehat{K}_B, \widehat{K}_B^{-1})$	$(\widehat{K}_C, \widehat{K}_C^{-1})$	$(\widehat{K}_O, \widehat{K}_O^{-1})$	
inner symmetric key	$\overline{K}_A$	$\overline{K}_B$	$\overline{K}_C$	$\overline{K}_O$	K_L
key share	$\sigma_\kappa(K)$				
medical data / anamnesis	φ_i				
pseudonym	ψ_{i_j}				
tags	τ_v				

TABLE I
DEFINITION OF SYSTEM ATTRIBUTES

of her data, and authorizing as well as revoking of all users is possible at any time, as for example defined by European Law. In case other legal acts demand that the health care provider should be the owner of the patient's data, the HCP could hold the root pseudonym on behalf of the patient. Moreover a rule could be added to our role-based access control model, which verifies, if a patient should be able to revoke rights of certain persons or institutions for a specific anamnesis.

In the following sections, we introduce the necessary operations and constraints to conduct the workflows of our approach. Furthermore, we show how authorizing and revoking of users works.

A. Establishing a Backup Keystore

We already mentioned that the need exists to assure that users still have access to their data if they lose their smart cards. In our system we provide an appropriate fall-back mechanism by sharing the user's inner private key, which grants access to the inner symmetric key and the pseudonyms. For instance, a relative could hold another encrypted version of the user's inner private key and, thus, would get the same rights as the user herself, if not controlled by a role-based access control model. Maybe someone does not want to grant access to all data to a certain relative. The demand arises to store a backup of the necessary keys inside the system. This would also ease recovering keys and issuing new smart cards. Due to security reasons, these secrets have to be divided between more persons.

In our prototype, we applied Shamir's threshold scheme [34] to divide the user's inner private key $\widehat{K}_A^{-1}$ into n shares $\sigma_\kappa(\widehat{K}_A^{-1})$. At least k of these n shares are necessary to reconstruct the whole key. The n shares are randomly distributed amongst all operators, which we therefore define as assigned operators. As any assigned operator may only hold a maximum of one share of a certain user's key, k necessary operators for every user exist that have to act together to unveil her key. This difference between n assigned and k necessary operators provides a fall-back mechanism and thus increases the availability of the system (cf. [30] for a security taxonomy) because one operator could be impeded from using his smart card. We named the set of assigned operators $\mathcal{O}^n \subset \mathcal{O}$ and the subset of necessary operators $\mathcal{O}^k \subseteq \mathcal{O}^n$. Following Shamir [34] it is not possible to combine $k-1$ shares to compute the key, but if an attacker is able to bribe $b \geq k$ operators, she may succeed in unveiling a certain user's identity. We state the probability of guessing the necessary operators for a specific user under the condition that the operators do not know for whom they are holding shares in equation (1), which is hypergeometrically distributed.

$$P(k \leq X \leq n) = \sum_{\kappa=k}^{n} \frac{\binom{n}{\kappa}\binom{|\mathcal{O}|-n}{b-\kappa}}{\binom{|\mathcal{O}|}{b}} \tag{1}$$

Using this equation leads to the following conclusions: (i) The larger the group of operators, the lower the probability that an attacker could bribe the assigned operators to find out a certain patient's identity. (ii) The lower the minimum of operators necessary to unveil the secret compared to the number of operators assigned to a certain patient, the higher the probability for a misuse of the system. If the operators do not know for which person they share secrets, an attacker has to compromise (in worst case) all operators minus the difference between the number of assigned and necessary operators to get access to a secret of a specific person.

In order to conceal the relation between operators and patients, the system encrypts the secret shares $\sigma_\kappa(\widehat{K}_A^{-1})$ with its logic key K_L and the inner public keys $\widehat{K}_O$ of the operators. As the operator can unveil the relation solely under the condition that she knows K_L, but still needs more operators to rebuild the shared secret, the possibility for arrangements between the operators is lowered.

In the upcoming section we provide the workflow of recovering a lost, destroyed or worn-out smart card.

B. Recover Lost Smart Card

With the defined constraints of the last paragraphs, we now present how recovering of a lost smart card works. Firstly, the user identifies against an operator and informs her to issue a new smart card. It is not mandatory, that the contacted operator has to hold a part of the inner private key of this user. Indeed she just starts the recovering process, by sending a message to the logic. The logic initiates a broadcasts to all operators $\mathcal{O}$ after encrypting the patient's identifier A_{id} with the logic key K_L. All operators look up their backup keystore by firstly encrypting the already encrypted patient's identifier with the particular operator's inner symmetric key $\overline{K}_O$ to query the

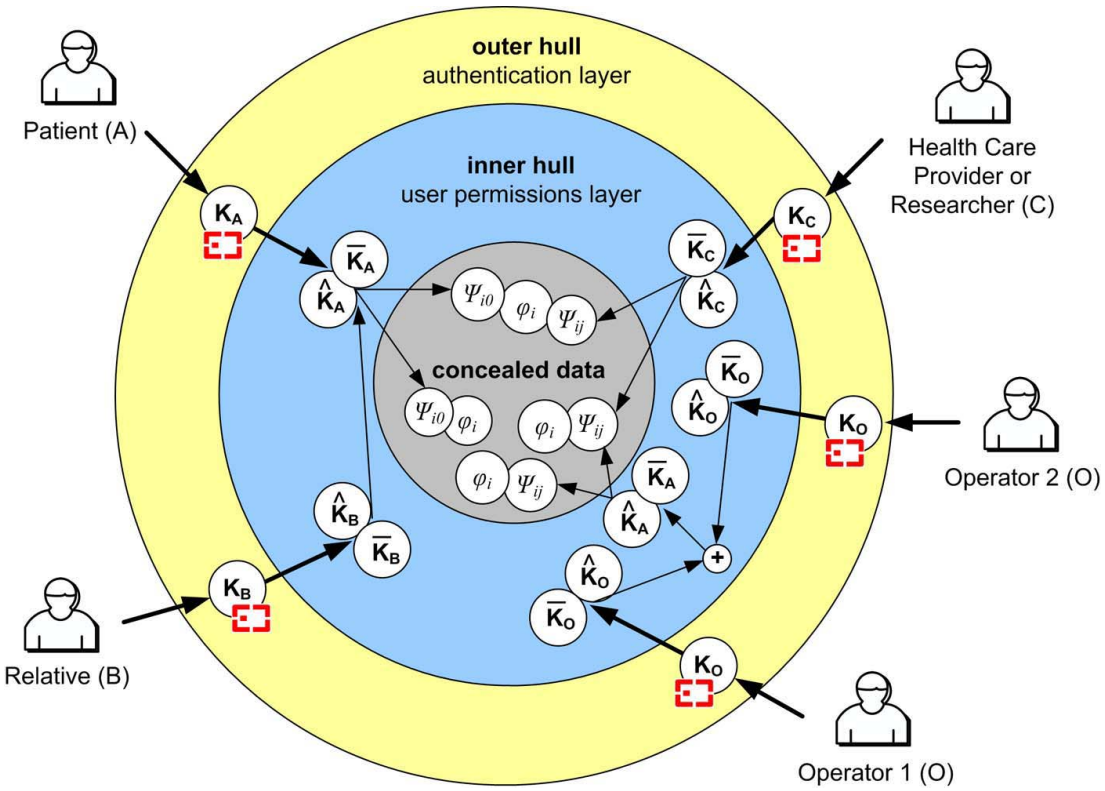

Fig. 1. PIPE security hull architecture

storage via the logic to see which one possesses a shared secret of the user's key. Any part which is available will be sent — after decrypting with the operators' symmetric key $\overline{K}_O$ — to the logic, which decrypts it with the logic key K_L and combines the partial keys. As soon as the logic received the shares from a minimum of k operators, the patient's inner private key can be re-constructed. Afterwards, a new outer public key pair will be created on the smart card and used to encrypt the inner private key. The logic updates this ciphertext in the database and issues the new smart card. The operators delete their shares and their relations to the user, because the logic randomly selects new operators to hold the shared secrets. At the same time, this workflow assures that the old smart card is not usable anymore.

On the one hand this backup-mechanism assures a very high level of security, but on the other hand obviously the costs for operating the whole system would increase as well, if all operators are human beings. Thus, we propose a combination of humans with smart cards and hardware security modules (HSM) [39] which, if operated as trusted instances and separated in different places, could act on behalf of human operators and still provide a reliable system.

C. Workflow Description

For information exchange between two or more actors, we use the notation of $i^{th}\ workflow\ step\ :\ Sender\ \rightarrow\ Receiver\ \rightarrow\ \ldots\ \rightarrow\ Receiver{:}\{Message\}$. If we apply a key as subscripted character, the message has been encrypted with this key. In the following, we state an example how the third message in a workflow between the patient and the logic, encapsulating the encrypted patient's identifier encrypted with the patient's inner symmetric key would look like.

Instance for a workflow step 3: $A \rightarrow L{:}\left\{\{A_{id}\}_{\overline{K}_A}\right\}$

In the next section we provide the basic workflows of our system. Firstly, we add actors to the system and set up relations between them. Afterwards we show the procedures for storing and retrieving of pseudonymized data.

D. Add Actor to the System

Every new actor in the system needs a smart card with her outer key pair stored on it, and a dataset in the identification database which includes the inner public key as well as encrypted versions of the inner symmetric and inner private key. Furthermore, the inner private key will be shared by $\mathcal{O}^n$ operators. As this workflow is identical to the one the relative has to go through and very similar to the one the health care provider conducts, we introduce the patient's workflow as an example.

1: $A \rightarrow L{:}\{A\}$

After identification against one or more persons (for example by applying the four-eye-principle), the patient is able

to send her personal data to the logic. The logic computes a unique identifier A_{id} for the patient.

Necessary operations: execute a hash-algorithm to compute the unique identifier A_{id}

2: $L \rightarrow St$:$\{\exists\ A_0 \in \mathcal{A} : A_{0_{id}} = A_{id}\ ?\}$

The logic looks up the storage to check if A_{id} already exists and hence verifies that the patient has not been added to the system yet.

Necessary operations: conduct one SQL select statement

3: $St \rightarrow L$:$\{\forall\ A \in \mathcal{A} : A_0\ \notin\ \mathcal{A}\ !\}$

After the storage replied with 'A_{id} unknown', the new identifier combined with the personal data is added to the identification database. Moreover, the logic retrieves the inner key pair as well as the inner symmetric key from the secured keystore of the trusted instance and starts the smart card production. The personalized smart card will hold the outer public key pair and is secured by a PIN.

Necessary operations: retrieve a key-pair and a symmetric key of secured keystore, initiate smart card production

4: $L \rightarrow St$:$\left\{\left\{\widehat{K}_A^{-1}\right\}_{K_A}, \left\{\overline{K}_A\right\}_{\widehat{K}_A}\right\}$

In the next step, the logic conducts the necessary encrypting operations and subsequently sends the inner private key, encrypted with the user's outer public key, as well as the inner symmetric key encrypted, with the user's inner public key, to the storage.

Necessary operations: encrypt two keys and execute one SQL insert statement

5: $L \rightarrow O$:$\left\{\left\{\left\{\sigma_\kappa(\widehat{K}_A^{-1}), A_{id}\right\}_{K_L}\right\}_{\widehat{K}_O}\right\} \forall\ \mathcal{O}^n$

The logic randomly selects $\mathcal{O}^n$ operators and uses the threshold scheme to divide the patient's inner private key into n shares. All shares are encrypted with the logic key K_L, subsequently encrypted with the inner public key $\widehat{K}_O$ of the particular operators and finally send to the operators. The logic encrypts the user's ID with the logic key as well as the particular operators key and transfers this ciphertext to the operators.

Necessary operations: apply threshold scheme, encrypt shares and patient's identifier twice for $\mathcal{O}^n$ operators

6: $O \rightarrow L \rightarrow St$:$\left\{\left\{\left\{\sigma_\kappa(\widehat{K}_A^{-1}), A_{id}\right\}_{K_L}\right\}_{\overline{K}_O}\right\} \forall\ \mathcal{O}^n$

The assigned operators decrypt the share and the user's identifier with their inner private keys $\widehat{K}_O^{-1}$. Then they encrypt both attributes again with their inner symmetric keys $\overline{K}_O$ and send these ciphertexts to the logic which forwards them to the storage.

Necessary operations: decrypt and encrypt the key shares and the user's identifier for $\mathcal{O}^n$ operators; $|\mathcal{O}^n|$ SQL insert statements to store the ciphertexts in the database

7: $L \rightarrow St$:$\left\{\widehat{K}_A\right\}$

The logic transfers the user's inner public key for posterior communication purposes to the storage and issues the new smart card with the outer key pair to the user.

Necessary operations: execute one SQL insert statement, finalize programming the smart card

In addition to the fact that the smart card programming and issuing component has to be a trusted instance, we established a role-based access control model, which controls that no operator is allowed to see the ciphertext of the patient's identifier encrypted with the logic key $\{A_{id}\}_{K_L}$ while processing (cf. step 6). Otherwise the operators could write down and compare the encrypted identifier with other operators and this could lead to frauds against the system.

E. Authorize User

A relation between two users is set up by exchanging their IDs and mutually encrypting them with their inner symmetric keys. This allows an user to create datasets for another user, which is controlled by a role-based access control model. Therefore, the logic provides the IDs of each participant encrypted with the particular public keys of the users. In order to describe the authentication process, we define function (2).

$$f_{authenticate}(U_{id}) := \begin{cases} \left\{\widehat{K}_U^{-1}\right\}_{K_U} & U_{id} \in St \\ errorcode & U_{id} \notin St \end{cases} \quad (2)$$

1: $U \rightarrow L$:$\{U_{id}\}$, $L \rightarrow St$:$\{f_{authenticate}(U_{id}) =?\}$, $St \rightarrow L \rightarrow U$:$\left\{\left\{\widehat{K}_U^{-1}\right\}_{K_U}\right\}$

The user U authenticates against her smart card by entering her PIN. If the PIN matches, the certificate of the client software is used to sign the user's identifier U_{id}. This signed identifier is transmitted to the logic which verifies this signature. If the certificate and the signature are valid, the logic queries the storage for the encrypted inner private key $\widehat{K}_U^{-1}$ of the specific user and forwards it to the user. She decrypts her inner private key with her outer private key K_U^{-1}. Therefore, even if the keystore in the client application has been successfully compromised, an attacker who gets access to the encrypted inner private key is not able to use it to gain access to the pseudonymized datasets until she does not have access to the outer private key, too. This authentication is as a pre-condition for all workflows including all participating users. Due to its similarity we leave out step 1 in the following sections.

Necessary operations: mutual authentication, one SQL select statement, decrypt inner private key

2a: $L \rightarrow C$:$\left\{\{A_{id}\}_{\widehat{K}_C}\right\}$

Firstly the logic encrypts the patients's identifier A_{id} with the health care provider's inner public key $\widehat{K}_C$ and transfers this ciphertext to the health care provider.

Necessary operations: one encrypt operation

2b: $L \rightarrow A$:$\left\{\{C_{id}\}_{\widehat{K}_A}\right\}$

Secondly the logic conducts the opposite operation with the health care provider's identifier C_{id} and the patient's inner public key $\widehat{K}_A$ and sends this ciphertext to the patient.

Necessary operations: one encrypt operation

3a: $C \rightarrow L \rightarrow St$:$\left\{\{\{A_{id}\},\{C_{id}\}\}_{\overline{K}_C}\right\}$

As the patient's identifier is encrypted for secure communication purposes, the health care provider needs to decrypt A_{id} with her inner private key $\widehat{K}_C^{-1}$ before she is able to apply her inner symmetric key $\overline{K}_C$. Moreover, she also encrypts her own identifier C_{id} with her inner symmetric key and transfers both attributes to the logic, which forwards it to the storage.

Necessary operations: decrypt and encrypt identifier, encrypt own identifier, one SQL insert statement

3b: $A \rightarrow L \rightarrow St$:$\left\{\{\{C_{id}\},\{A_{id}\}\}_{\overline{K}_A}\right\}$

Compared to step 3a the patient conducts the mirrored operations. The health care provider's identifier will be decrypted with the patient's inner private key and afterwards encrypted again with the patient's inner symmetric key. The patient also encrypts her identifier and sends both to the logic which inserts both ciphertexts in the database via the storage.

Necessary operations: decrypt and encrypt identifier, encrypt own identifier, one SQL insert statement

To revoke an authorization (e.g., for a health care provider), the patient deletes the certain entry from the relation table by querying the ciphertexts. Afterwards our role-based access control model — the top layer of the security stack — denies the right of adding new datasets for the certain patient-health care provider combination.

Authorizing and revoking of the relation between a relative B and a patient A is similar to the workflow between a health care provider and a patient. Additionally, the patient provides the relative via the logic with her inner private key, which the relative stores encrypted with her inner symmetric key. This means, that the relative would have the same rights as the patient, if we would not use a role-based access control model to control the relatives view on the data. In our prototype the relative is only allowed to read all data, writing and authorizing is denied.

In the prior sections we introduced the workflows of how users can be added to the system and showed the set-up of relationships between them. With this basis we can consequently add pseudonymized data to the system.

F. Add Anamnesis to System

All medical data in our approach is separated from the identification data to assure users' privacy. Therefore, all datasets are associated with j pseudonyms. Every pseudonym is unique for any patient-health care provider-anamnesis combination. Thus a so-called root pseudonym ψ_{i_0} exists, which is only related with the patient and the i^{th} anamnesis dataset.

2a: $L \rightarrow C$:$\left\{\{\{A_{id}\},\{\psi_{i_j}\}\}_{\widehat{K}_C}\right\}$

2b: $L \rightarrow A$:$\left\{\{\{C_{id}\},\{\psi_{i_0}\},\{\psi_{i_j}\}\}_{\widehat{K}_A}\right\}$

The logic encrypts the patient's identifier with the health care provider's inner public key and sends it to the health care provider. Afterwards the logic conducts the opposite encrypt operations for the patient with the health care provider's identifier and the patient's inner public key. Furthermore, the logic generates two new random numbers, which represent the root pseudonym ψ_{i_0} and the pseudonym ψ_{i_1}, which will be shared between the health care provider and the patient. The logic encrypts these pseudonyms with the particular inner public keys and sends the ciphertexts to the health care provider and the patient.

Necessary operations: encrypt identifier, generate two new unique random numbers, one encrypt operation for the patient related pseudonym and two encrypt operations for the pseudonym, which will be shared between the patient and the health care provider

3a: $C \rightarrow L$:$\left\{\{\{A_{id}\},\{C_{id}\},\{\psi_{i_j}\},\{\tau_v\}\}_{\overline{K}_C},\{\varphi_i\}\right\}$

The health care provider firstly decrypts the patient's identifier and the pseudonym ψ_{i_j} with her inner private key. Secondly, she begins to form the message for adding the new anamnesis by appending the anamnesis data in plain-text. Afterwards, she encrypts the pseudonym, the patient's and the health care provider's identifier as well as the related chosen tags τ_v with the health care provider's inner symmetric key. Finally, she transmits this message to the logic.

Every participant on an anamnesis may hold different tags, in other words a keyword like x-ray or surgery. Consequently, if two health care providers are authorized to access a certain anamnesis, they may apply tags that differ from each other. As this descriptive information of the dataset needs to be hidden to avoid data mining and thus prevent guessing of a patient's identity, tags and other identifiers are stored encrypted with the particular users' inner symmetric keys. To assure an appropriate runtime of the system, every SQL query is only conducted with ciphertexts to minimize the encrypting operations of metadata. Even if it is possible to select anamnesis by the identifiers of the participants, the tags have to be chosen carefully to achieve more appropriate results on the retrieval of datasets and in order to optimize the runtime. Moreover, the anamnesis' timestamps need to be hidden. We propagate splitting up timestamps into their atoms. For example May 02, 2007 can be basically split into the tags May, 02 and 2007. Moreover, this was a $Wednesday$, which is another tag. If we also add the week of the year, in our example the $week\_14$, time ranges become queryable as well. A practical example would be a radiologist who invites patients for a check-up a week after the surgery. To receive a list of all patients who have been screened in week 13 and need to re-attend a week later, she forms a query of her encrypted identifier and of the tags $knee$, $x-ray$, $needs\_follow\_up$, $Thursday$ and $week\_13$. The result will be the pseudonyms and their related metadata,

which are both encrypted with the health care providers inner symmetric keys. Afterwards it is possible for her to decrypt the related metadata, select the desired anamnesis datasets and retrieve the data as described in the next section.

Necessary operations: decrypt identifier and pseudonym, encrypt tags, patient as well as health care provider identifiers and pseudonym

3b: $A \rightarrow L$:$\left\{\{\{A_{id}\},\{C_{id}\},\{\psi_{i_0}\},\{\psi_{i_1}\},\{\tau_v\}\}_{\overline{K}_A}\right\}$

The patient decrypts her opposite's identifier and both pseudonyms ψ_{i_j}, ψ_{i_0} with her inner private key. Afterwards she sends the related chosen tags, the patient's as well as the health care provider's identifier and the pseudonyms, all encrypted with the patient's inner symmetric key, to the logic.

Necessary operations: decrypt identifier and pseudonyms, encrypt tags, patient as well as health care provider identifiers and pseudonyms

4: $L \rightarrow St$:$\left\{\{\psi_{i_j},\tau_v,A_{id},C_{id}\}_{\overline{K}_C}\right\}$
5: $L \rightarrow St$:$\left\{\{\psi_{i_0},\psi_{i_j},\tau_v,A_{id},C_{id}\}_{\overline{K}_A}\right\}$
6: $L \rightarrow St$:$\{\psi_{i_0},\psi_{i_j},\varphi_i\}$

The logic transfers the anamnesis data φ_i and a plain-text version of the pseudonyms to the anamnesis database in the storage. Afterwards the logic saves the encrypted tags, patient's and health care provider's identifiers as well as the encrypted pseudonyms in the pseudonyms table of the storage on which the users may afterwards execute a SQL select statement with the ciphertexts of exemplarily the patient's identifier and her chosen tags.

Necessary operations: three insert statements

G. Retrieve Anamnesis from System

There are different ways to retrieve an anamnesis. First of all the patient or a relative who have access to the patient's inner private key $\widehat{K}_A^{-1}$ are able to decrypt the patient's inner symmetric key. A health care provider who has been authorized for a certain anamnesis may access the medical data with the appliance of her inner symmetric key $\widehat{K}_A^{-1}$. Hence, all of these users are able to query the storage via the logic by encrypting the necessary tags, like keywords or a time-stamp in combination with an encrypted version of the identifiers, to look up a certain anamnesis with a SQL select statement. Moreover, it is possible to hand over a couple of pseudonymized datasets to a research institution. We present the workflow of a patient who wants to see a specific anamnesis.

2: $A \rightarrow L \rightarrow St$:$\left\{\{\tau_v,A_{id},C_{id}\}_{\overline{K}_A}\right\}$

The patient prepares the where clause in the SQL statement by encrypting the chosen tags, for example a keyword, timestamp or health care provider's identifier. The patient transfers the query to the storage via the logic.

Necessary operations: encrypt patient's identifier and desired tags

3: $St \rightarrow L \rightarrow A$:$\left\{\{\psi_{i_0}\}_{\overline{K}_A}\right\}$ $for\ |\psi_{i_0}| \geq 1$

If the query produced any results, the storage replies with a minimum of one or a set of encrypted root pseudonyms which the logic forwards to the patient.

Necessary operations: one SQL select statement

4: $A \rightarrow L \rightarrow St$:$\{\psi_{i_0}\}$ $for\ |\psi_{i_0}| \geq 1$
5: $St \rightarrow L \rightarrow A$:$\{\varphi_i\}$

The patient selects from the received list of pseudonyms and decrypts the desired pseudonym/s with her inner symmetric key and queries the logic with the plain-text pseudonym/s. The logic forwards the patient's request to the storage. The storage returns the matching anamnesis via the logic. Additionally the logic provides the patient with all related tags of a certain pseudonym, even if they have not been within the query.

Necessary operations: decrypt and encrypt ψ_{i_0}, one SQL select statement for every desired anamnesis

As all datasets are already pseudonymized it is also possible to conduct secondary usage of the stored medical data with this workflow. If no relation between more datasets is required (e.g. to show the clinical history of a certain patient), the researchers are authorized by adding another pseudonym, which the researcher and the patient share. This is comparable to the authorization of a health care provider for an anamnesis, but without the exchange of the patient's identifier, which will not be handed over to the researcher. In case it is necessary to base the results of a study on the medical history of the patients, it is also possible to use the same tag for a series of anamnesis. If the researchers want to invite the patients to a follow-up study or if new information about the studies' outcomings are available, they can add a flag to the specific anamnesis. This flag will be shown, for example in form of a dialog window, the next time a health care provider, authorized for this anamnesis, or the patient herself authenticates against the system.

IV. Conclusions

The implementation of electronic health records does not only promises a higher level of service quality for the patients, but also reduces costs for social insurance systems and therefore for the society. As highly sensitive data is stored and handled in nation-wide medical systems, there is the requirement for assuring the patients' privacy to avoid misuse. Although several approaches for managing anamnesis systems exist, their underlying security is too weak to assure confidentiality of life-long medical data storage. Moreover people need to be convinced of such centralized systems as they are strongly concerned about their privacy.

In this paper we discussed a variety of existing approaches and their security shortcomings, such as their dependence on a centralized patient-pseudonyms list, a life-long pseudonym or the concealment of an algorithm. We worked out several principles with the focus on assuring the confidentiality, integrity, availability and privacy of sensitive patient-related

medical data. Furthermore, we contributed a secure and efficient architecture for the combined primary and secondary usage of health-related data based on these principles. Our system PIPE assures that the patient is in full control of her data with the maximum of gainable security, achieved by applying authorization on encryption [20], in- and outside the system as well as for all communication. In other words, even if all communication between the actors is transmitted over unsecure channels like the Internet, the confidentiality is granted because all attributes in the database are already secured by encryption. For integrity purposes, we additionally propose the usage of Transport Layer Security (TLS) or signed messages. As users possess smart cards as security tokens in our approach, we introduced a secure fall-back mechanism if a smart card has been lost, stolen, compromised or just worn out. We defined the administrative role operator which is in charge to hold a backup of the user keys in their keystore. Moreover we applied a threshold scheme [34] to securely divide the backup keys between the operators to assure inner system's security.

V. Further Work

During the last months we began to conduct several load tests with our prototype. The testbed consist of ≈ 100.000 $patients$ and $\approx$ 10.000 $health\ care\ providers$ which hold together ≈ 500.000 $anamnesis\ datasets$. The relations between the patient, health care providers and anamnesis have been set-up randomly and are approximately equally distributed. As we did not have sufficient resources to produce a unique smart card for each user, we decided to base our test on 100 smart cards by using one smart card for several actors. However, the load tests we conducted show that this restriction of the prototype does not have any influence on the performance results. PIPE is nearly linear scalable because only encrypted keys and random numbers have been applied to realize the hull structure and therefore the performance heavily depends on the underlying database and server machine. Although, the performance overhead of encrypting on the smart card's cryptochip may be further lowered, if the user's inner private key as well as the user's inner symmetric key would be cached on the smart card, the results are very promising and allow the deployment of the system in case studies with partner companies. Further work will provide more statistical and economical insights on the applied threshold scheme. Moreover, we will present a workflow for ad-hoc access to a subset of the anamnesis data, the patient's emergency data (cf. [40]). With this publication we introduce our realization of an additional access routine for emergency doctors. In order to refine the usage of our system, we will publish the details of our role-based access control model, which operates as the top-layer of our stacked security approach.

VI. Acknowledgment

We want to thank our master students Mathias Kolb and Markus Pehaim as well as the members of our business partner Braincon Technologies, Oswald Boehm, Alexander Krumboeck, and Gert Reinauer for their support, further Stefan Jakoubi for his review.

This work was performed at Secure Business Austria, a competence center that is funded by the Austrian Federal Ministry of Economics and Labor (BMWA) as well as by the provincial government of Vienna.

References

[1] S. Maerkle, K. Koechy, R. Tschirley, and H. U. Lemke, "The PREPaRe system – Patient Oriented Access to the Personal Electronic Medical Record," in *Proceedings of Computer Assisted Radiology and Surgery, Netherlands*, 2001, pp. 849–854.

[2] F. R. Ernst and A. J. Grizzle, "Drug-related morbidity and mortality: Updating the cost-of-illness model," University of Arizona, Tech. Rep., 1995.

[3] ——, "Drug-related morbidity and mortality: Updating the cost-of-illness model," University of Arizona, Tech. Rep., 2001.

[4] J. Pope, "Implementing EHRs requires a shift in thinking. PHRs–the building blocks of EHRs–may be the quickest path to the fulfillment of disease management." *Health Management Technology*, vol. 27(6), p. 24, 2006.

[5] United States Department of Health & Human Service, "Hipaa administrative simplification: Enforcement; final rule," *Federal Register / Rules and Regulations*, vol. Vol. 71, No. 32, 2006.

[6] European Union, "Directive 95/46/ec of the european parliament and of the council of 24 october 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data." *Official Journal of the European Communities*, vol. L 281, pp. 31–50, 1995, http://europa.eu/scadplus/leg/en/lvb/l14012.htm.

[7] Council of Europe, *European Convention on Human Rights*. Martinus Nijhoff Publishers, 1987.

[8] Republic of Austria, "Datenschutzgesetz 2000 (DSG 2000), BGBl. I Nr. 165/1999," 1999.

[9] M. Ackerman, R. Craft, F. Ferrante, M. Kratz, S. Mandil, and H. Sapci, "Telemedicine technology," *Telemedicine Journal and e-Health*, vol. 8, No. 1, pp. 71 –78, 2002.

[10] J. Montagnat, F. Bellet, H. Benoit-Cattin, V. Breton, L. Brunie, H. Duque, Y. Legr, I. E. Magnin, L. Maigne, S. Miguet, J. M. Pierson, L. Seitz, and T. Tweed, "Medical images simulation, storage, and processing on the european datagrid testbed," *Journal of Grid Computing*, vol. 2, Number 4, pp. 387–400, 2004.

[11] A. Pfitzmann and M. Koehntopp., "Anonymity, Unlinkability, Unobservability, Pseudonymity, and Identity Management A Consolidated Proposal for Terminology," in *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2005.

[12] K. Taipale, "Technology, Security and Privacy: The Fear of Frankenstein, the Mythology of Privacy and the Lessons of King Ludd," *International Journal of Communications Law & Policy*, vol. 9, 2004.

[13] R. L. Peterson, "Patent: Encryption system for allowing immediate universal access to medical records while maintaining complete patient control over privacy," *US Patent US 2003/0074564 A1*, 2003.

[14] C. Thielscher, M. Gottfried, S. Umbreit, F. Boegner, J. Haack, and N. Schroeders, "Patent: Data processing system for patient data," *Int. Patent, WO 03/034294 A2*, 2005.

[15] G. de Moor, B. Claerhout, and F. de Meyer, "Privacy enhancing technologies: the key to secure communication and management of clinical and genomic data," *Methods of information in medicine*, vol. 42, pp. 148–153, 2003.

[16] J. Gulcher, K. Kristjansson, H. Gudbjartsson, K., and Stefanson, "Protection of privacy by third-party encryption in genetic research," *European journal of human genetics*, vol. 8, pp. 739–742, 2000.

[17] K. Pommerening, "Medical Requirements for Data Protection," in *Proceedings of IFIP Congress, Vol. 2*, 1994, pp. 533–540. [Online]. Available: citeseer.ist.psu.edu/330589.html

[18] K. Pommerening and M. Reng, *Medical And Care Compunetics 1*. IOS Press, 2004, ch. Secondary use of the Electronic Health Record via pseudonymisation, pp. 441–446.

[19] D. Lobach and D. Detmer, "Research challenges for electronic health records," *American Journal of Preventive Medicine*, vol. 32, Issue 5, pp. 104–111, 2007.

[20] B. Riedl, T. Neubauer, G. Goluch, O. Boehm, G. Reinauer, and A. Krumboeck, "A secure architecture for the pseudonymization of medical data," in *Proceedings of the Second International Conference on Availability, Reliability and Security*, 2007, pp. 318–324.

[21] A. Rector, J. Rogers, A. Taweel, D. Ingram, D. Kalra, J. Milan, P. Singleton, R. Gaizauskas, M. Hepple, D. Scott, and R. Power, "Clef - joining up healthcare with clinical and post-genomic research," in *Proceedings of UK e-Science All Hands Meeting*, 2003, pp. 203–211. [Online]. Available: citeseer.ist.psu.edu/rector03clef.html

[22] A. Lysyanskaya, R. L. Rivest, A. Sahai, and S. Wolf, "Pseudonym systems," in *Proceedings of the Sixth Annual Workshop on Selected Areas in Cryptography (SAC '99)*. [Online]. Available: citeseer.ist.psu.edu/lysyanskaya99pseudonym.html

[23] U. Flegel, "Pseudonymizing unix log files," in *Proceedings of the International Conference on Infrastructure Security*. London, UK: Springer-Verlag, 2002, pp. 162–179.

[24] J. Daemen and V. Rijmen, *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer; 1 Edition, 2002.

[25] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[26] V. S. Miller, "Use of elliptic curves in cryptography," *Lecture notes in computer sciences*, vol. 218 on Advances in cryptology—CRYPTO 85, pp. 417–426, 1986.

[27] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. Wiley; 2 edition, 1995.

[28] M. Hendry, *Smart Card Security and Applications, Second Edition*. Norwood, MA, USA: Artech House, Inc., 2001.

[29] W. Rankl and W. Effing, *Smart Card Handbook*. New York, NY, USA: John Wiley & Sons, Inc., 1997.

[30] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 11–33, 2004.

[31] R. Russell, D. Kaminsky, R. F. Puppy, J. Grand, D. Ahmad, H. Flynn, I. Dubrawsky, S. W. Manzuik, and R. Permeh, *Hack Proofing Your Network (Second Edition)*. Syngress Publishing, 2002.

[32] T. Westran, M. Mack, and R. Enbody, "The last line of defense: a host-based, real-time, kernel-level intrusion detection system," in *submitted to IEEE Symposium on Security and Privacy*, 2003.

[33] T. Thornburgh, "Social engineering: the "Dark Art"," in *Proceedings of the 1st annual conference on Information security curriculum development*. New York, NY, USA: ACM Press, 2004, pp. 133–135.

[34] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[35] J. Han and M. Kamber, *Data mining: concepts and techniques*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000.

[36] K. Maris, "The Human Factor," in *Proceedings of Hack.lu, Luxembourg*, 2005.

[37] V. Schmidt, W. Striebel, H. Prihoda, M. Becker, and G. D. Lijzer, "Patent: Verfahren zum be- oder verarbeiten von daten," *German Patent, DE 199 25 910 A1*, 2001.

[38] B. Riedl, V. Grascher, and T. Neubauer, "Pseudonymization for securing e-health applications," in *to appear in the proceedings of the 13th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC07)*, 2007.

[39] D. C. Wherry, "Secure your public key infrastructure with hardware security modules," SANS Institute, Tech. Rep., 2003.

[40] B. Riedl and O. Jorns, "Granting access to emergency data in a pseudonymized e-health architecture," in *submitted to the Proceedings of the 9th International Conference on Information Integration and Web-based Application & Services*, 2007.