Applying a Threshold Scheme to the Pseudonymization of Health Data

Bernhard Riedl, Veronika Grascher, Thomas Neubauer
Secure Business Austria, Vienna
riedl, grascher, neubauer@securityresearch.ac.at

Abstract

Due to the cost pressure on the health care system an increase in the need for electronic healthcare records (EHR) could be observed in the last decade because EHRs promise massive savings by digitizing and centrally providing medical data. As highly sensitive patient information is exchanged and stored within such a system, legitimate concerns about the privacy of the stored data occur, as the lifelong storage of medical data is a promising target for attackers. These concerns and the lack of existing approaches that provide a sufficient level of security raise the need for a system that guarantees data privacy and keeps the access to health data under strict control of the patient. This paper introduces PIPE (Pseudonymization of Information for Privacy in e-Health), a new EHR architecture for primary and secondary usage of health data. PIPE's security model is based on pseudonymization instead of encryption.

1 Introduction

The availability of sound information is essential for health care providers' decisions regarding the patients' care and thus for the quality of treatment and patients' health [4]. Therefore, the idea of nation-wide electronic health records (EHR) has been introduced within the past several years as a method for improving communication and collaboration between health care providers. On the one hand, implementing EHRs promises massive savings by digitizing medical data like diagnostic tests and images [8]. On the other hand, research groups can benefit from the disclosure of anamnesis data for R&D reasons. Although a centralized storage could decrease the operational costs of the medical care system, patients are concerned about their privacy. For instance, a history about substance abuse or HIV infection could result in discrimination or harassment.

In this paper, we introduce a new system for the pseudonymization of health data that differs from existing approaches in its ability to securely integrate primary and secondary usage of health data (cf. [3, 6, 7] for a description of primary and secondary use) and, thus, provides a solution to security shortcomings of proposed systems. This paper especially focuses on the description of an administrative role that holds a backup of the users' keys and, thus, provides a secure fall-back mechanism, if a smart card has been lost, stolen, compromised or just worn out. Compared to existing approaches, our concept does not depend on a patient list, which reflects the association between a patient's identification and medical data or a breakable algorithm. Instead, we base our architecture PIPE (Pseudonymization of Information for Privacy in e-Health) on a layered structure that guarantees that the patient is in full control of her data. This concept can be used as an extension to EHR applications but also as basis for national EHR initiatives.

2 System Description

The set of users $\mathcal{U}$ in our system PIPE is divided into the roles patient, relative, health care provider and operator. The data owner - as demanded by many legal acts (cf. [2, 9]) - is the patient ($A \in \mathcal{A}$), who is in full control of her datasets. Every patient may permit one or more relatives ($B \in \mathcal{B}$) to access all of her anamnesis data. Health care providers ($C \in \mathcal{C}$) are another instance which can be authorized to read or append a subset of the patient's medical data.

	User	*Operator*
abbreviation	U	O
unique identifier	U_{id}	O_{id}
outer key pair	(K_U, K_U^{-1})	(K_O, K_O^{-1})
inner key pair	$(\widehat{K}_U, \widehat{K}_U^{-1})$	$(\widehat{K}_O, \widehat{K}_O^{-1})$
inner symmetric key	$\overline{K}_U$	$\overline{K}_O$
key share	$\sigma_\iota(K)$	
anamnesis	φ_i	
pseudonym	ψ_{i_j}	

Table 1. Definition of System Attributes

Moreover, administrative roles, which we named opera-

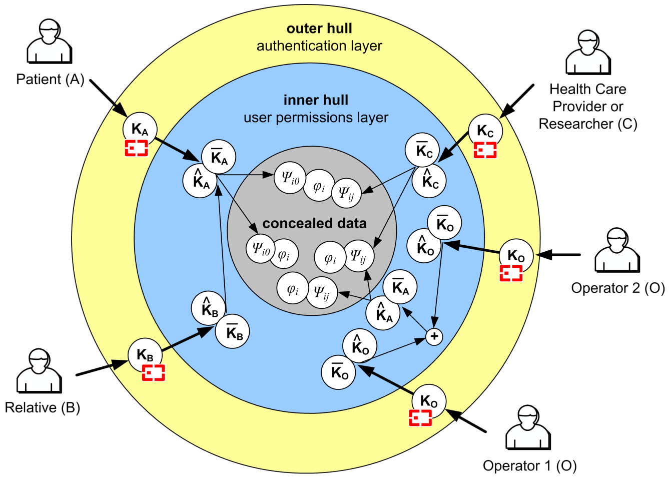

Figure 1. PIPE security hull architecture

tors ($O \in \mathcal{O}$), exist for the purpose of system maintenance. As we use smart cards in PIPE, these operators share the secrets of the patients to establish a back-up mechanism for worn-out, destroyed, compromised or lost smart cards. Table 1 provides an overview of the cryptographic keys and used abbreviations of our approach. The notation K stands for a key, K^{-1} for a private key. For example the inner private key of an operator is defined as $\widehat{K}_O^{-1}$. The identification as well as the anamnesis data is held in the storage St, which further acts as the secured keystore. Figure 1 provides an overview of the PIPE's security hull architecture (cf. [10]), which is controlled by a central logic module. The users' secrets are distributed amongst the different hulls. In other words, every hull is made of one or more users' keys or hidden relations between users, other users and their data. The secret of a specific hull can solely be accessed with the appliance of the plain-text secrets from the next outer hull. The anamnesis datasets φ_i, which are each identified with j pseudonyms ψ_{i_j}, can only be read or changed by using the inner symmetric key $\overline{K}_U$.

$$\left\{\left\{\left\{\psi_{i_j} \mapsto \varphi_i\right\}_{\overline{K}_U}\right\}_{\widehat{K}_U}\right\}_{K_U} \qquad (1)$$

We give an example for the encryption sequence for an user's data in equation (1). The user's inner private key $\widehat{K}_U^{-1}$ in the inner hull — or user permissions layer of the user U — has been previously encrypted with the user's outer public key K_U, which is stored on her smart card. The smart card, which itself is secured with a PIN code, is the security token of the outer hull or authentication layer. Moreover, the inner symmetric key $\overline{K}_U$ is held encrypted with the inner public key. Therefore, if a user wants to read her anamnesis data, she first of all decrypts her inner public key with her outer private key after authenticating against her smart card. Secondly, she uses her inner private key to decrypt her inner symmetric key. With the appliance of her inner symmetric key she consequently gets access to her medical data via the encrypted pseudonyms.

Pipe's hull architecture allows two different types of authorization. On the one hand, users may equip someone (e.g., a relative) with full access to their data by sharing their inner private key. In that case, the relatives store the user's inner private key encrypted with their inner public key. On the other hand, it is possible to permit access for health care providers to specific anamnesis by appending pseudonyms to datasets. To avoid data mining in the storage, pseudonyms are unique for any patient-health care provider-anamnesis combination and thus prevents the creation of patient's profiles.

In our opinion, this approach can be considered secure, but lacks a fall-back mechanism, if a patient for instance looses her smart card. Hence, there is the need to assure that a system's participant still gets access to her datasets in such a case.

3 The Secure Backup Keystore

As the users must be provided with a backup mechanism in case they do not possess a working smart card, we introduce a new solution to securely backup the user's inner private key. To ease restoring backup keys, we propose the usage of a centralized key storage, accessible for the operators. One possibility to assure confidentiality of this backup key storage is to use a role-based access control model to assign and control the operators' permissions. As role-based access control models can be compromised, by-passed or the administrative users may suffer, for example, a socialengineering attack [5], the demand arises to encrypt the keystore itself [10] and to share the responsibility of restoring a certain user's key between several persons. Therefore, we applied Shamir's threshold scheme [11] together with encryption in our prototype to avoid misuse of the users' keys and consequently of the users' data.

All users' inner private keys $\widehat{K}_U^{-1}$ are automatically divided into n shares $\sigma_\iota(\widehat{K}_U^{-1})$ upon creation. These shares are distributed randomly and independently amongst the operators. We define the set of operators assigned to hold a part of a certain user's key $\mathcal{O}^n \subset \mathcal{O}$ and the subset of operators necessary to unveil a certain user's key $\mathcal{O}^k \subseteq \mathcal{O}^n$. The delta between the number of assigned operators and necessary operators may be seen as backup operators, because one operator could not be available for immediate recoverage of a certain user's key. Following Shamir [11], it is not possible to compute the key with $k-1$ shares, but if an attacker is able to bribe $b \geq k$ operators, she may succeed in unveiling a certain user's identity. Equation (2) states the probability of guessing at least k operators to reconstruct the secret for a specific user under the condition that the operators do not know for whom they are holding shares. This probability is hypergeometrically distributed.

$$P(k \leq X \leq n) = \sum_{\iota=k}^{n} \frac{\binom{n}{\iota}\binom{|\mathcal{O}|-n}{b-\iota}}{\binom{|\mathcal{O}|}{b}} \qquad (2)$$

Equation (2) is based on the constraint that the operators do not know (i) the identity of the users for whom they hold secret shares $\sigma_\iota(\widehat{K}_U^{-1})$ and (ii) which operators are their counterparts to calculate the whole key. We concealed the association between the operators and a certain user by providing our logic module with a symmetric key K_L, which will be firstly used to encrypt the identifiers and the secret shares, before the operators apply their keys to veil the secrets. Only with knowledge of K_L, it is possible for the operators to find out for which user they have been assigned, but she still need more operators to rebuild the shared secret.

In order to rebuild a lost smart card with access to the user's inner private key, the user identifies against an operator. The following section introduces the workflow of recovering a lost key.

1: $O \rightarrow L{:}\{U_{id}\}$

It is not necessary that this operator has to hold a part of this user's inner private key. In fact, she just initiates the recovering process by sending a message to the logic module.

Necessary operations: proof user's identity

2: $L \rightarrow O{:}\left\{\{U_{id}\}_{K_L}\right\} \forall\, \mathcal{O}$

The central logic module broadcasts a message to all operators $\mathcal{O}$ with an encrypted version of the user's identifier U_{id} because, as mentioned in the previous section, the logic module key K_L has been used to envelope the identifier first.

Necessary operations: encrypt user's identifier

3: $O \rightarrow L \rightarrow St{:}\left\{\left\{\{U_{id}\}_{K_L}\right\}_{\overline{K_O}}\right\} \forall\, \mathcal{O}$

Upon receipt, all operators query their backup keystore via the central logic module by encrypting these ciphertexts with the particular operator's inner symmetric key $\overline{K}_O$. With this message the logic module is able to find out which operator possesses a user's key share.

Necessary operations: encrypt shares by $|\mathcal{O}|$ operators

4: $St \rightarrow L \rightarrow O{:}\left\{\left\{\left\{\sigma_\iota(\widehat{K}_U^{-1})\right\}_{K_L}\right\}_{\overline{K_O}}\right\} \forall\, \mathcal{O}$

After querying the double encrypted ciphertexts against the storage, the logic module receives associated double encrypted key shares and forwards them to the assigned operators.

Necessary operations: $|\mathcal{O}|$ SQL select statements

5: $O \rightarrow L{:}\left\{\left\{\sigma_\iota(\widehat{K}_U^{-1})\right\}_{K_L}\right\} \forall\, \mathcal{O}^n$

The next step is that all assigned operators decrypt their particular shared secrets with their inner symmetric key $\overline{K}_O$ and transmit it to the logic module. The logic module is now able to decrypt these shares with its key K_L and consequently to combine the parts. As soon as the logic module receives the shares from a minimum number of k necessary operators, the user's inner private key can be re-calculated by applying Shamir's threshold scheme (cf. [11]).

Necessary operations: decrypt a maximum of $|\mathcal{O}^n|$ key shares, apply threshold scheme

6: $L \rightarrow St{:}\left\{\left\{\widehat{K}_U^{-1}\right\}_{K_{U'}}\right\}$

Afterwards, the logic module retrieves a new outer key pair $(K_{U'}, K_{U'}^{-1})$ from the storage which will replace the outer keys (K_U, K_U^{-1}) of the lost smart card. The logic module uses the new outer public key to encrypt the user's inner private key. The logic module saves this ciphertext in the storage and initiates the smart card production. To avoid replay-attacks the storage moreover deletes the operator shares and their relations to the user.

Necessary operations: generate new asymmetric key pair, encrypt user's inner private key

7: $L \rightarrow O{:}\left\{\left\{\left\{\sigma_\iota(\widehat{K}_U^{-1}), U_{id}\right\}_{K_L}\right\}_{\widehat{K}_O}\right\} \forall\, \mathcal{O}^n$

Subsequently, the logic module randomly chooses $\mathcal{O}^n$ assigned operators and uses the threshold scheme to divide the user's inner private key into n shares. Once more, all shares will be double-enveloped. Firstly, the logic module applies its key K_L and secondly, encrypts the gained ciphertexts with the certain inner public keys $\widehat{K}_O$ of the selected operators. These encrypted secret shares will then be transmitted to the operators. Moreover, the logic module applies the same encryption procedures to the user's ID U_{id} and transfers this ciphertext to the operators, too.

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

$$8\text{: } O \to L \to St\text{:}\left\{\left\{\left\{\sigma_\iota(\widehat{K}_U^{-1}), U_{id}\right\}_{K_L}\right\}_{\overline{K}_O}\right\} \forall\, \mathcal{O}^n$$

Upon receipt, the assigned operators decrypt their particular shares 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 return these ciphertexts to the logic module which saves them in 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

With this workflow, we provide the recovery scenario for lost or destroyed smart cards. We assure that after a strong identification task, the user's smart card can be replaced. Moreover this scenario may be used to quickly lock compromised smart cards.

4 Conclusions and Further Work

Electronic health records do not only promise a significant reduction of the costs for managing medical information, they also achieve a higher level of service quality for patients [1]. As highly sensitive data is stored and handled in nation-wide medical systems, there is an increasing demand for assuring the patients' privacy in order to avoid misuse. Although several approaches for the realization of EHRs exist, the security of these systems is often too weak to assure confidentiality of life-long medical data storage. This especially holds for their dependence on a centralized patient-pseudonyms list, a life-long pseudonym or the concealment of an algorithm. Based on these shortcomings, we introduced the secure architecture PIPE for both primary and secondary usage of health-related data. As users in our system possess smart cards as security tokens, this papers especially focused on introducing a secure fall-back mechanism, if a smart card has been lost, stolen, compromised or just worn out. Therefore, we proposed an administrative role who holds a backup of the users' keys. Moreover, we applied a threshold scheme based on Shamir's secret sharing to securely divide the backup keys between the operators in order to assure inner system's security. Our system assures that a patient is in full control of her data with the maximum of gainable security, achieved by applying authorization on encryption, in- and outside the system as well as for all communication.

Acknowledgment

We want to thank the members of our business partner Braincon Technologies, Oswald Boehm, Alexander Krumboeck, and Gert Reinauer for their support and 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] F. R. Ernst and A. J. Grizzle. Drug-related morbidity and mortality: Updating the cost-of-illness model. Technical report, University of Arizona, 2001.

[2] 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*, L 281:31–50, 1995. http://europa.eu/scadplus/leg/en/lvb/l14012.htm.

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

[4] 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*, pages 849–854, 2001.

[5] K. Maris. The Human Factor. In *Proceedings of Hack.lu, Luxembourg*, 2005.

[6] K. Pommerening. Medical Requirements for Data Protection. In *Proceedings of IFIP Congress, Vol. 2*, pages 533–540, 1994.

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

[8] 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*, 27(6):24, 2006.

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

[10] 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*, pages 318–324, 2007.

[11] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.