FAKULTÄT FÜR !INFORMATIK

Economic and Security Aspects of Pseudonymization Approaches in eHealth

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering & Internet Computing

eingereicht von

Mathias Kolb, BSc.
Matrikelnummer 0327557

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung:
Betreuer: o.Univ.Prof. Dipl.-Ing. Dr. A Min Tjoa
Mitwirkung: Dipl.-Ing. Mag. Dr. Thomas Neubauer

Wien, 09.06.2009

(Unterschrift Verfasser) (Unterschrift Betreuer)

Erklärung der Selbstständigkeit

Mathias Kolb, Römerweg 3/7, 3430 Tulln

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen - die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, den 09.06.2009 Mathias Kolb

Kurzfassung

Vor kurzem hat im Gesundheitswesen das digitale Zeitalter begonnen und neue Datenschutzprobleme erzeugt. Datenschutz ist ein Grundrecht jedes Menschen. Regierungen versuchen dieses durch Gesetze zu stärken. Vor allem im Gesundheitsbereich spielt der Datenschutz eine bedeutende Rolle, weil mit vielen sensiblen Daten gearbeitet wird. Zur Unterstützung des Datenschutzes wurden Sicherheitslösungen, z.B. Pseudonymisierung, entworfen, die aber oft nicht mit den Gesetzen konform gehen. Diese Konzepte beinhalten Zutrittsmechanismen, welche im Falle eines Verlustes oder Defektes wiederhergestellt werden müssen. Leider sind diese Backup-Systeme oftmals ineffizient und mangelhaft.

Diese Diplomarbeit analysiert sechs verschiedene Pseudonymisierungsanwendungen und evaluiert ihre Schwachstellen im Bezug auf ihr Konzept bei Verlust der Karte. Außerdem werden die Anwendungen auf die Einhaltung der rechtlichen Vorgaben von der Europäischen Union und den Vereinigten Staaten von Amerika überprüft.

Auf der Basis der Pseudonymisierungsanwendungen PIPE - Pseudonymization of Information for Privacy in e-Health - wird in dieser Diplomarbeit eine Verbesserung des Schlüssel-Backup-Systems vorgestellt. Es wird ein zweiteiliges Schlüssel-Backup-System entwickelt, welches mit Hilfe von Hardware Security Modulen und unter Verwendung von Shamir's threshold scheme arbeitet. Dadurch konnte die Sicherheit ohne große Leistungseinbußen erhöht werden. Ein Beispiel zeigt, dass, verglichen mit dem originalen Backup-System, die durchschnittliche Laufzeit des Wiederherstellungsprozesses nur ca. 2% länger dauert.

Diese Diplomarbeit soll die Entwicklung zukünftiger Pseudonymisierungsanwendungen durch Auflisten von rechtlichen und technischen Anforderungen erleichtern und ein angemessenes und sicheres Backup-System garantieren.

Abstract

Recently, the digital era has begun in the health care sector and as a matter of fact this sector is faced with new privacy issues. As privacy is a fundamental right of every individual, several laws were enacted that demand the protection of patients' privacy. On the basis of these laws, pseudonymization concepts were developed, which help to protect the privacy of each individual. However, these pseudonymization concepts often do not comply with legal or basic security requirements. As these pseudonymization concepts are based on tokens needed for accessing the system, they have to provide a fall-back mechanism in case the tokens are lost, stolen or worn out. Unfortunately, these fall-back mechanisms are often inefficient and inadequate.

This master thesis analyzes six different pseudonymization approaches and evaluates their major drawbacks, especially regarding drawbacks in their fall-back mechanisms. Moreover, the compliance of those approaches with the applicable legal situations in the European Union and the United States of these six pseudonymization approaches is investigated.

Based on the pseudonymization concept PIPE - Pseudonymization of Information for Privacy in e-Health - this master thesis outlines improvements for that fall-back mechanism. A two-folded key sharing system has been developed, which uses Hardware Security Modules and Shamir's threshold scheme. The security has been enhanced with less of an impact on the performance. As an example shows, the average execution time takes only 2% longer compared to the original fall-back meachanism.

This thesis presents contributions to future developments of pseudonymization approaches by providing legal and technical requirements and an adequate and secure fall-back mechanism.

Acknowledgments

I would like to thank o.Univ.Prof. Dipl.-Ing. Dr. A Min Tjoa for providing me with an opportunity to grow as a student and engineer in his research area.

I also thank Dr. Thomas Neubauer and Dr. Bernhard Riedl for technical and non-technical advice, support, and editing.

Special thanks to Veronika Grascher for her mathematical input and review and to the team of Secure Business Austria for their help.

This work is dedicated to Norbert Kolb and Kriemhild Kolb. They always motivated me during my education and contributed to it.

Table of Content

CHAPTER 1

Introduction

This chapter gives an introduction of the master thesis. This chapter primarily answers the questions, ‘Why is pseudonymization important?’ and ‘What are the goals to be solved?’

1.1 Motivation

Since some years, there has been a revolution in the administration of the health care sector. As medical treatment of patients is extremely expensive and time consuming, more and more electronic health record (EHR) systems are being introduced. Such systems have to protect patients’ privacy on the one hand and improve efficiency and reduce costs on the other hand. EHRs may improve communication between health care providers and the access to medical documents, which leads to better quality service by health care providers [24]. EHR systems promise cost reduction by digitally storing medical documents and images and could save the health care sector $77.8 billion annually (5%) in the United States, if they were widely used [63]. Furthermore, EHR systems could not only reduce costs, they could also prevent adverse drug events, which cause more than 200,000 cases of death a year in the United States [11]. In fact, these systems could support health care providers with guidelines for the administration of medical drugs.

In addition, electronic health record systems could process a large amount of medical data. As the need of society is to interconnect such systems to improve the efficiency, for example, over the Internet, the risk of abusive use of this highly sensitive data will gain in importance. It is a fundamental right of every individual to demand privacy, because the disclosure of sensitive data may cause serious problems for them. For example, insurance companies or employers could use this information to deny a health service contract or employment.

To protect the privacy of the patient's medical data, several different approaches are currently still under development. Most of them use a pseudonymization technique to divide the medical data into an identifying and anonymous part. These pseudonymization techniques often use a security token, which stores secret keys or unique identifiers. During the process of dividing the data, a pseudonym is generated with the help of the keys or identifiers, which are stored on this security token. If this security token is lost, stolen or becomes worn out, the data will be lost forever, because the divided data can only be found and combined with the secret keys or identifiers on the security token. Therefore, optimal approaches have to provide adequate mechanisms to recover such a security token.

To overcome this issue, some approaches use a patient list, on which the pseudonyms are linked to patients. Such a patient list is used by the approaches of Pommerening [35] or by the approach of Thielscher [58]. Other pseudonymization systems like Elektronische Gesundheits Karte (eGK) [3, 6–9, 16, 29, 43, 65] use a second security token to relink the data. If the first security token has been lost or stolen, the data will be recovered with the second security token and a new security token will be created. These recovery techniques have major drawbacks, because an insider could abuse such a patient mapping list or in the case that both security tokens are destroyed, stolen or lost, all the data would be lost forever.

To protect the abusive use of the recovery mechanism, PIPE (Pseudonymization of Information for Privacy in e-Health) [37–41] introduces a new distributed security token recovery mechanism. The secret keys or identifiers are shared with several operators of the system by using Shamir's threshold scheme [49].

1.2 Goals and Contributions

- Which pseudonymization approaches adhere to the current privacy laws?

As already mentioned, it is important to protect the privacy of an individual's personal data, such as medical data, for example, x-rays, blood reports, etc. Therefore, some pseudonymization approaches will be presented and their privacy protection mechanism will be compared to current privacy laws.

- What are the major drawbacks of pseudonymization approaches?
- How can the safe recovery of data be guaranteed if the data has been encrypted and the patient has lost her security token?

To answer these two questions, six approaches (PIPE, EGK, Approaches of Pommerening, Approach of Thielscher, Approach of Peterson, Approach of Slamanig and Stingl) will be analyzed against drawbacks, especially drawbacks in the security token recovery mechanism. To be able to investigate this recovery mechanism, an introduction will be given first.

- Are there possible improvements of PIPE?
- Is the improved fall-back mechanism of PIPE economically justifiable?
- Is the performance of PIPE's improved fall-back mechanism fast enough?

The goal is to improve the fall-back mechanism of PIPE and to show that this improvement is economically justifiable and also fast enough what is being realized. The economic question will be answered by providing formulas to calculate the costs of the system. In this thesis, the costs of the improved fall-back mechanism for about 50 million users will be presented. The performance analysis will be done by presenting the speed of the functions of the original and improved fall-back mechanism in O-Notation and also by calculating the execution time of these functions on the base of the 50 million users example.

1.3 Structure of the Thesis

As the focus of this master thesis is the investigation of different pseudonymization approaches; the legal requirements for such approaches must first be stated. Therefore, privacy laws in the European Union, such as the Data Protection Directive (95/46/EC) [12] and in the United States, such as the Health Insurance Portability and Accountability Act of 1996 (HIPAA) [59–61] will be presented. To conclude which pseudonymization approaches adhere to the current laws, legal requirements will be examined from these acts. Furthermore, some additional functional requirements will be presented.

To be able to compare these requirements with different pseudonymization approaches, an overview will be given and their security drawbacks will be presented. Moreover, as most of these approaches use a security token for data pseudonymization, a fall-back mechanism has to exist, which is able to recover a lost, stolen or worn out security token. The investigation of such a fall-back mechanism will be a part of this thesis.

After the privacy laws in the European Union and United States have been introduced and an overview of existing pseudonymization approaches has been given, an evaluation of the current implemented legal and functional requirements will be stated.

Next, the core workflows of Pseudonymization of Information for Privacy in e-Health (PIPE) [37–41] - one of the invested approaches - will follow, and furthermore, the fall-back mechanism of this approach will be extended. Finally, the costs and performance of the new introduced fall-back mechanism will be analyzed. The performance of the fall-back mechanism is given in O-Notation and the execution times are calculated based on an example.

CHAPTER 2

Privacy laws in the European Union and United States

This chapter gives an overview of privacy protection laws in the European Union (EU) and United Stats (U.S.). It pays special attention to privacy protection laws, which apply to the health care sector.

2.1 Facts about Privacy

Nowadays, society is collecting all kinds of information. In daily life, several types of information, which could be quite sensitive or damaging to individuals and organizations, are tracked [44, 46, 47]. For example, the supermarket tracks which items have been bought, mobile phone providers keep track of customer movements, airlines know what type of seat and meal is preferred and hotel chains keep records of room preferences. Moreover, the exchange and storage of this information could be done extremely cheaply and easily over the Internet. Whether the collected data may only be disclosed to third parties in a de-identifiable form, for example, by deleting the name and social insurance number, but nevertheless, there could still be enough data to be able to do a data mining process. So, de-identifiable data does not guarantee anonymity [44, 56]. Sweeney presented an example in [57], where medical data is combined with an electronic version of a city's voter list, which was purchased for only twenty dollars.

For this reason it is more important than ever to protect the privacy of an individual. In more than 30 countries, privacy laws protect the data of individuals [22]. The content of these privacy laws varies in each country, but they are mostly based on the Organization for Economic Cooperation and Development (OECD) Guidelines on the Protection of Privacy and Transborder Flows of Personal Data [32].

The next sections will give an overview about the privacy laws in the European Union and United States.

2.1.1 European Union

Throughout history, collected information of individuals has been abused in several ways. For example, in the years during World War II, the German government abused census data to identify households of certain ethnic, religious or other targeted groups [50]. They deliberately annihilated approximately 6 million European Jews and other 'undesirable' population groups such as Gypsies, homosexuals, and various categories of the disabled during the Holocaust [48].

As various states gained in power and size, the first privacy laws were introduced against this danger. The first national data-protection law was passed 1973 in Sweden, followed by the United States in 1974 and West Germany in 1977 [5]. By the end of the seventies more and more European States had passed privacy laws. To spread these laws across Europe, the European Union (EU) passed the Data Protection Directive (95/46/EC) [12] in 1995. This directive applies to all personal data, which is collected or processed either electronically or in old-fashioned paper-filing systems. Article 2(a) of the Data Protection Directive (95/46/EC) [12] defines personal data as:

> '... any information relating to an identified or identifiable natural person (data subject); an identifiable person is one who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his physical, physiological, mental, economic, cultural or social identity ...'

Moreover, the Data Protection Directive (95/46/EC) is based on eight principles to which all data controllers are subject. These principles limit the usage of collected personal data [12, 13, 19]:

1. The data must be processed fairly and lawfully.
2. The data must be collected for explicit and legitimate purposes and used accordingly.
3. The data must be accurate and where necessary, kept up to date.
4. Organizations have to provide mechanisms to correct, delete or block data.
5. The data that identifies individuals must not be kept longer than necessary.
6. The data must be processed in accordance with the rights of the data subject.
7. Every organization must ensure the security and integrity of personal data, that they are processing.
8. It is not permitted to transfer personal data outside the European Union unless the country ensures an adequate level of protection

Furthermore, to ensure fair and lawful processing of the collected data, the data controller has to inform the data subjects which data will be collected and used concerning them. The individuals also must be informed of which types of third parties the collected data will be disclosed to, and the data subject must have the possibility to decline [12, 13, 19, 52].

2.1.2 United States

In the United States (U.S.), privacy has not gained much political attention. Discussions on privacy have been driven often by events in Europe. In the 1970s, concerns over privacy reached new heights, because there had been abuses of wiretapping powers, tax, bank and telephone records during the Watergate scandal [50]. These concerns gave birth to the Privacy Act of 1974, which applies only to records of personal information held by federal agencies. These agencies are allowed to keep records only if relevant and necessary. They are not allowed to create secret files of an individual without giving the right to copy their own files. Furthermore, agencies are not permitted to disclose these records without the agreement of the individual - except within the agency for routine use or to law enforcement [50]. To protect private electronic communications from unauthorized access by the government, the Electronic Communications Privacy Act of 1986 and the Computer Matching and Privacy Protection Act of 1988 have been introduced. In addition to these laws, no general privacy act currently exists, which could be compared with the European approach. There are a handful of laws which cover the use of private data in health care [59–61], the electronic commerce industry [21], the cable-television industry [62] and a few other areas.

A definition of personal data is given in Section 8(8) of the Online Privacy Protection Act (OPPA) [21]:

> '... information collected online from an individual that identifies that individual, including first and last name; home and other physical address; e-mail address; social security number; telephone number; any other identifier that the Commission determines identifies an individual; or information that is maintained with, or can be searched or retrieved by means of, data described above ...'

2.2 Privacy in eHealth

As already mentioned, nowadays privacy protection has become extremely important. Especially sensitive data like in the health care sector need more privacy protection than non-sensitive data in other sectors. Sensitive medical data like the state of medical health, for example being HIV positive or having chronic illness, could harm a person if they are accessed by unauthorized persons. For example, an employer who accesses medical data unauthorized of her employees, could use this information to dismiss an employee. Another example could be of an insurance company denying a contract because of a chronic illness. Therefore the focus will be on privacy laws, which are related to the health care sector in this section.

2.2.1 European Union

In the European Union, the Data Protection Directive (95/46/EC) [12] already implements protection for sensitive data, which are related to racial and ethnic background, political affiliation, religious or philosophical beliefs, trade-union membership, sexual preferences and health [12, 13, 19, 50]. Besides this Data Protection Directive (95/46/EC) [12], an additional Working Document [14] has been released by the Article 29 Working Party of the European Union, which provides guidelines for the interpretation of the data protection legal framework for EHR systems and explains some of the general principles. The Working Document also gives indications on the data protection requirements for setting up EHR systems, as well as for the applicable safeguards. The processing of sensitive data is generally prohibited but is tolerated under specific circumstances [13]. Some of these circumstances are:

- if the data subject explicitly agrees on the processing of her sensitive data.
- if the processing of data is allowed by law.
- if the subject is unable to agree on the processing because of being disabled in an accident, for example a car accident.

Unlike non-sensitive data, sensitive data could only be disclosed if the data subject would give consent. Furthermore the data controller will need to be registered for this purpose. On the contrary, anonymized sensitive data, for example, medical data, would fall outside of the Data Protection Directive as long as the individual cannot be identified. The Secretary of State for Health could declare anonymized data as unanonymized to bring it back into the Directive [19]. Until now, no such declaration has been made.

Furthermore the Protection Directive (95/46/EC) defines the rights for the individual. Some of these rights are:

- to receive information about the processing of their own data
- to receive a copy of all personal data held by the data controller
- the prevention of direct marketing and automated decision-making

- to seek damages for breach of the data protection principles

Germany

In Germany, the government passed a law (§291a SGB V) which defines the structural and functional requirements of an electronic health record infrastructure [20]. This law includes a list of obligatory and optional services (ref. table 2.1) which should be implemented. In addition, access permissions of different actors, for example of health care providers, doctors, physiotherapist, etc. are well-defined. Furthermore, regulations exist to prevent fraudulent use of data that is stored in the electronic health record system.

obligatory	*optional*	*additional*
insurance data	emergency data	medication supervision
e-prescription	pharmaceutical documentation	reference cases
	electronic medical report	
	electronic health record	
	general patient data	

Table 2.1: Germany: §291a SGB V - obligatory and optional services

2.2.2 United States

In the United States, as already mentioned, a privacy protection law for medical data exists. The Health Insurance Portability and Accountability Act of 1996 (HIPAA) [1, 4, 10, 59–61] is based on five principles: Consumer Control, Boundaries, Accountability, Public Responsibility and Security. The five principles only apply to individually identifiable health information, which is [19, 59–61]:

- created by or received from health care provider, health plan, employer or clearinghouse.
- related to the provision of health care or the past, present or future medical condition.
- identifies or could reasonably be used to identify an individual.
- has been transmitted electronically or maintained in any other form or medium.

However, the act does not include other medical data, for example car insurance that has medical coverage or general sickness absence in the workplace that is not the subject of the health plan [19].

The disclosure of Protected Health Information (PHI) is permitted in certain cases. For example, the data is disclosed to the individual self, the data is de-identified or to carry out health plan's own treatment, payment or health care operations. Furthermore the data owner could give consent to the processing of her medical data. Moreover, medical data could be disclosed to the clergy, for example, by the hospital, unless the patient declines.

To protect the privacy of individuals, many rights have been set up under the Health Insurance Portability and Accountability Act. Individuals have the right:

- to inspect or copy their own information
- to request amendment or correction of erroneous or incomplete information
- to request the restriction of use or disclosure
- to give authorization for certain uses and disclosures

CHAPTER 3

Evaluation of common pseudonymization approaches

In the last few years, electronic health record systems have gained more and more importance. One fact for the increasing demand of EHR systems, is that these systems could help to reduce costs and improve the quality of health care. For example, widespread use of EHR systems could save the health care system $77.8 billion annually (5%) in the United States [63]. This cost reduction is done, for example by improved exchange of medical data between health care providers or hospitals and by reduction of paper wastage. Furthermore, EHR systems could reduce adverse drug events (ADR), which causes more than 200,000 cases of death a year in the United States [11]. In fact, the systems could support health care providers with guidelines for the administration of medical drugs. Moreover, EHR systems could improve research by providing more appropriate data that allow monitoring of diseases and help to develop new medication or treatment methods [39].

3.1 Overview of EHR approaches

Currently, three different EHR approaches are in use. Firstly, there is th plain-text approach in which all data is readable for everyone. This approach could be compared with the traditional paper-record system. Secondly, there is the encrypted-text approach in which all data are encrypted and only accessible to persons with the key to decrypt this data. Thirdly, there is the pseudonymization approach, in which only the reference between the data and the data owner is encrypted. Table 3.1 shows an short overview of the approaches, which will be described in this chapter and will be grouped into the three categories mentioned above.

Description	*Name*	*References*
Plain-text approach	Approaches of Pommerening	[35]
Encrypted-text	Approach of Peterson	[33]
	Elektronische Gesundheits Karte	[3, 6–9, 16, 29, 43, 65]
Pseudonymization approach	Pseudonymization of Information for Privacy in e-Health	[37–41]
	Approach of Thielscher	[58]
	Approach of Slamanig and Stingl	[51, 53, 54]

Table 3.1: Overview of EHR approaches

3.2 Pseudonymization of Information for Privacy in e-Health (PIPE)

PIPE (Pseudonymization of Information for Privacy in e-Health) [37–41] is an approach developed by Secure Business Austria in cooperation with Braincon Technologies. This approach introduces a new form of architecture, which provides the following contributions:

- authorization of health care providers or relatives to access defined medical data on encryption level.
- secure fall-back mechanism in case the security token is lost or worn out.
- the data are stored without the threat of data profiling.
- secondary use without establishing a link between the data and the owner.

Furthermore, PIPE [37–41] is based on a server-client architecture as shown in figure 3.1.

The client is a small service, which provides an interface to legacy applications, manages requests to local smart card readers and creates a secure connection to the server. The server, also called Logic (L), handles requests from clients to the storage. The data in the

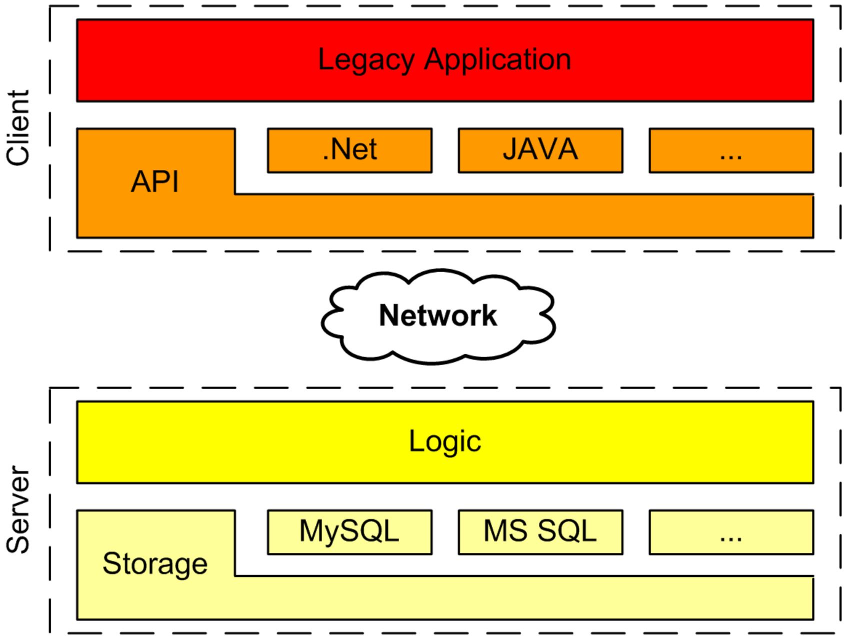


Figure 3.1: PIPE: System Overview

storage is divided into two parts, the personal data and the pseudonymized medical data. As shown in figure 3.2, the link between personal data and pseudonymized medical data is protected through a hull-architecture. Furthermore, the hull-architecture contains a minimum of three security-layers: the authentication layer (outer hull), the user permission layer (inner hull) and the concealed data layer. To reach the next hull, there are one or more secrets, for example, symmetric or asymmetric keys or hidden relations in every hull-layer. A definition of all system attributes can be found in table 3.2.

Moreover, PIPE contains users which have different roles, for example, patient A, relative B, health care provider C or operator O. The patient is the owner of her data and has full control of her datasets. She is able to view her medical data, add and revoke health care providers and she may define relatives, who have the same rights as herself. Health care providers can be authorized to see and create subsets of anamnesis data by the patient. The operators are the administrators of the system.

- The authentication layer contains an asymmetric key pair, for example the patient outer public key K_A and outer private key K_A^{-1}. These keys are stored on a smart

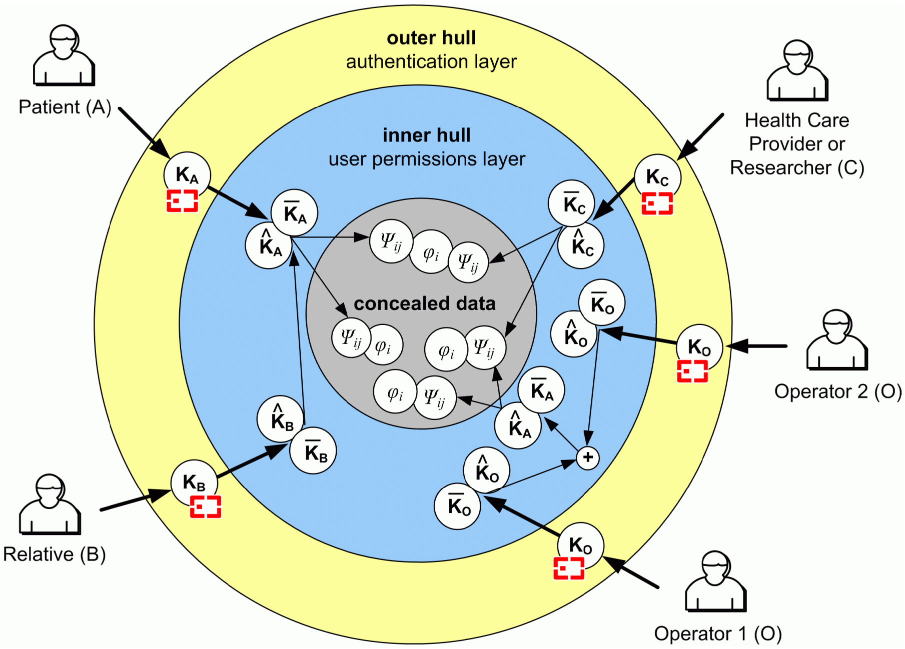


Figure 3.2: PIPE: Layered model representing the authorization mechanism [37–41]

card and are protected with a pin code. The outer private key is used to decrypt the keys of the permission hull-layer.

- The permission layer contains an asymmetric key pair and a symmetric key, for example the patient inner public key $\widehat{K_A}$, inner private key $\widehat{K_A}^{-1}$ and symmetric key $\overline{K}_A$. The symmetric key is encrypted with the inner private key and is used to en-/decrypt pseudonyms in the concealed data layer. If a patient associates a relative, her inner private key $\widehat{K_A}^{-1}$ will be encrypted with the relatives inner public key $\widehat{K_B}$. So, the relative is able to decrypt the patient's symmetric key $\overline{K}_A$ with her inner private key $\widehat{K_B}^{-1}$ until the patient's inner private key $\widehat{K_A}^{-1}$ is changed.
- The concealed data layer contains hidden relations which are called pseudonyms. Each medical data set is associated with one or more pseudonyms ψ_{i_j}. As the patient is the owner of her medical data and the person with security clearance, she owns the so called root-pseudonym ψ_{i_0} [41]. These pseudonyms are calculated with an algorithm, which is based on a secret key. In this case, this secret key is the symmetric key of the user. Only users, who are able to decrypt one of these

pseudonyms ψ_{i_j} can rebuild the link between the patient and her medical data.

To find the pseudonyms to rebuild the link to the medical data, the authors introduced keywords. These keywords are selected on creation time of the medical data or when another user is authorized. These keywords are encrypted with the symmetric key of the root user and the user who will be authorized. After the keywords are stored in the database, the user can select any of these keywords to find the pseudonym. Every additional keyword restricts the result more.

	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})	(K_L, K_L^{-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	$\overline{K}_A$	$\overline{K}_B$	$\overline{K}_C$	$\overline{K}_O$	$\overline{K}_L$
key share				$\sigma_\iota(K)$	
medical data / anamnesis	φ_i				
pseudonym	ψ_{i_j}				

Table 3.2: PIPE: Definition of System Attributes

Fall-back mechanism

The hull-architecture assures a strong security system, but if, for example, a patient loses her smart card or the smart card is worn out, all data would be lost forever. Hence, PIPE implements a fall-back mechanism to replace the smart card. Therefore, operators O have been introduced who share the inner private key $\widehat{K}_A^{-1}$ of a patient. To decrease the risk of abuse only several operators based on the four-eye-principle could re-build the key. Therefore, the patient's inner private key $\widehat{K}_A^{-1}$ is divided into shared secrets by the use of Shamir's threshold scheme [49]. This scheme allows sharing keys between several operators. The key is shared with N_A ($N_A \subset N$) randomly assigned operators and to recover the key, N_k ($N_k \subseteq N_A$) operators are needed. Additionally, operators have no knowledge of which keys they hold.

Security issues

PIPE is designed to implement a high level of security. The fact that the medical data is not encrypted to enable the option for secondary use may lead to a security issue. An attacker who breaks into the database could commit a data profiling attack if the unencrypted medical data contains any identifying words.

Another attack could be a social engineering attack, where the attacker fakes the identity of the person she wants to attack. Additionally, the attacker has to fake an official photo identification like a passport or a driver's license and request a new smart card for the system. This form of attack has a low risk, because there are mechanisms to prevent such an attack. For example, the new smart card could be sent by registered mail to the principal residence and only handed over to the owner. Additionally, the pin code for the new smart card will be sent by mail after the owner obtained the new smart card.

3.3 Elektronische Gesundheits Karte (eGK)

The electronic health card (elektronische Gesundheits Karte - eGK) [3, 6–9, 16, 29, 43, 65] is an approach of the Fraunhofer Institute supported by the Federal Ministry of Health Germany. Since the beginning of 2007, a field study has been started in selected cities of Germany. The goal of this study is to get the system ready to introduce the system all over the country.

EGK is designed as a service-oriented architecture (SOA) with some restrictions. One of these restrictions is that the health card could only be accessed locally on the client's side. Another restriction is that services should use remote procedure calls for communication due to performance and availability issues. Therefore, the system architecture is split up into five layers [9].

1. The *presentation* layer defines interfaces to communicate with the user.
2. The *business logic* layer combines different services which are processed automatically.
3. The *service* layer provides special functional uncoupled services.

4. The *application* layer primarily realizes the user right and data management.
5. The *infrastructure* layer contains all physical hardware and software management, for example, data storage, system management, virtual private networks, etc.

With this layered architecture, the system has the possibility to provide several service applications. Some of these are emergency data, electronic prescription, electronic medical report or an electronic health record system [20]. Furthermore, the eGK system can be extended with additional applications [29]. Therefore, the Frauenhofer Institute invented a ticketing concept to implement some uncoupled actions in combination with security mechanisms to comply with the privacy policy [8, 20]:

Hybrid encryption: All data which will be stored in the virtual file system is encrypted with a one-time symmetric key, called a session key. This session key is encrypted with the public key of the patient. To decrypt the data, the patient has to decrypt the session key with her private key, and finally, the data will be decrypted with this session key.

Authentication: A user is authenticated by using the Challenge-Response technique. Therefore, the system generates a random number. This number will be encrypted with the public key of the user. Only the user has the possibility to decrypt this random number with the private key, which is stored on her health card and can send it back to the eGK system.

Furthermore, the ticketing concept manages the access rights to the system. Therefore, a virtual file system has been introduced as shown in figure 3.3. A file or directory in this virtual file system has a default ticket-toolkit and an unlimited number of private ticket-toolkits, called t-node in figure 3.3. A user can define a private ticket-toolkit for every other user in the system. This private ticket-toolkit could have stronger or looser access policies as the default ticket-toolkit. The ticket-toolkit contains a ticket-building tool, a ticket-verifier, the access policy list and an encrypted link to the directory or file. Every user holds a root directory in the virtual file system, which does not have a parent node. Furthermore, any directory contains unencrypted links to the ticket-toolkits of their child nodes. This technique enables the system to perform a fast selection of sub nodes (select * from t-nodes where parentID = directoryID) [8, 16].

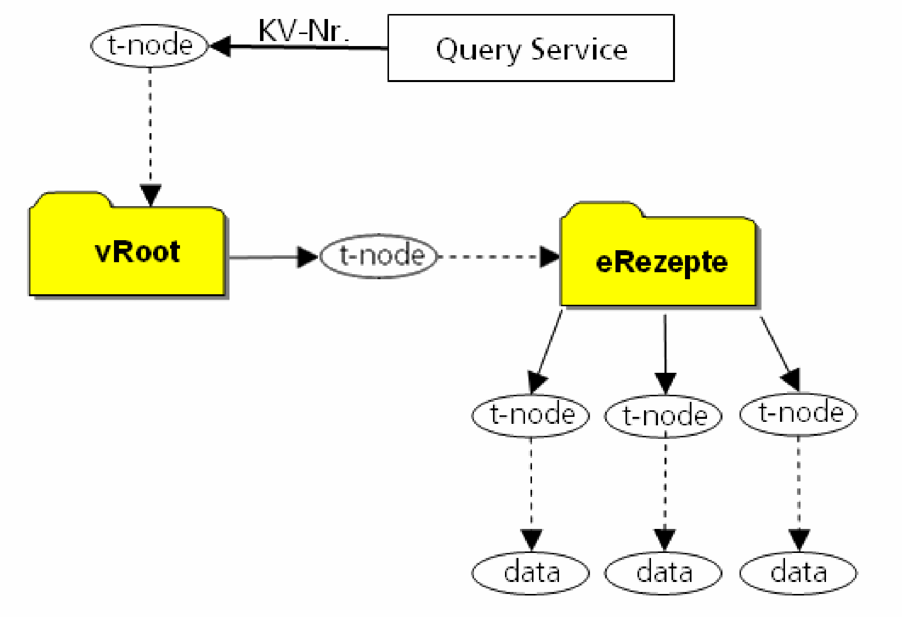


Figure 3.3: eGK: Virtual filesystem [16]

To be able to find the root node of a specific user, a query service has been introduced. The query service maps a unique identifier, for example the insurance number to the internal user and returns a ticket-toolkit containing an encrypted link to the root node. If there is no private ticket-toolkit available for the user who performed the request, the system returns a default ticket-toolkit, which is based on a challenge. If the user is able to solve this challenge, she will have the access rights, which have been defined in the default access policy.

Fall-back mechanism

Both the hybrid encryption and the challenge response technique are based on the asymmetric key pair, which is stored on the health card of the patient. There are not any possibilities to backup these keys elsewhere. In the case that the patient loses the smart card or the card becomes worn out, the data would be lost forever. Neither the operating company nor any public administration organization [8] could ever recover the data, which has been stored in the system.

To overcome this problem, the eGK architecture provides an option with the possibility to store a second private ticket-toolkit for every entry. This private ticket-toolkit uses a asymmetric key pair, which is stored on an emergency card. The architecture does not specify this emergency card. There are several possibilities, for example, to use the card of a family member as an emergency card [8, 16].

In case the card has been lost, the patient requests a new health card. Therefore, the emergency card is used to decrypt the session keys of the second ticket-toolkit, and finally, these session keys are encrypted with the keys of the new health card. Additionally, a new second private ticket-toolkit will be created for the new emergency card. After this process is completed, the system won't accept the old health and emergency cards.

Security issues

The eGK provides a high level of security. Even if an attacker breaks into the database, she would not be able to link and read the data stored there. An attacker could perhaps attempt a data profiling attack and retrieve some information from the unencrypted keywords, if these contain any identifiable words.

3.4 Approaches of Pommerening

Pommerening [35] proposes different approaches for secondary use of electronic health records (EHR). Therefore, he differs between one-way and reversible pseudonyms. In the following paragraphs, an overview about some of these approaches will be given.

The first approach is based on data from overlapping sources for one-time secondary use. In this case, overlapping sources could be, for example, data from different EHRs or for probes from a biomaterial bank or data, which have been collected on another examination. To connect the data, a unique identifier (PID) has been introduced. Figure 3.4 shows the pseudonymization workflow. Therefore, a pseudonymization service encrypts the PID with a hash algorithm, and the medical data has been encrypted with the public key of the secondary user. This secondary user can decrypt the medical data and merge the data of a person, but cannot identify it.

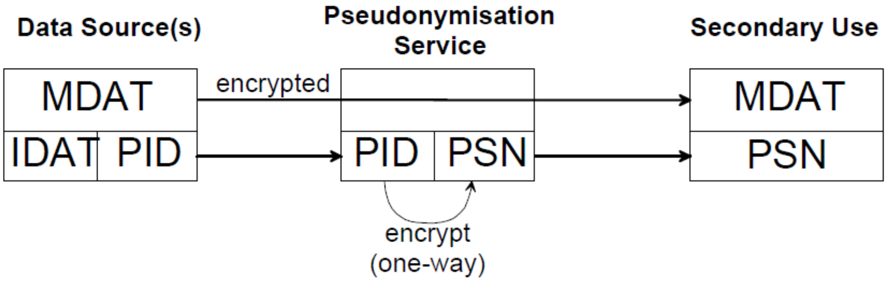


Figure 3.4: Pommerening: Data Flow for One-Time Secondary Use [35]

The second approach is also based on one-time secondary use, but with the possibility to re-identify the patient. Therefore, Pommerening extends the first approach with a PID service, which stores a reference list containing the identity of the patient and the associated PIDs. In case the patient should be notified, the pseudonymization service decrypts the pseudonym and sends the request to the PID service, which has the possibility to notify the data source owner.

The third approach fits the need of a research network with numerous secondary usages, and it also supports long-term observation of a patient with, for example, chronic diseases. It also allows to send research results to the patient or her responsible health care provider. The export and pseudonymization procedure is shown in figure 3.5. Therefore a physician exports her local database to the central researcher database. The identification data will be replaced with a PID in the PID service. For each secondary use the data will be exported through the pseudonymization service. The PID is encrypted by the pseudonymization service with a project specific key to ensure that different projects get different pseudonyms.

Security issues

The approaches of Pommerening have a serious drawback. The generated pseudonyms from the PID service are stored in a reference patient list to be able to re-build the link to the patient. To enhance the security, this list will be stored at a third party institution, but this measure does not prevent an abuse of the list through an insider of the third

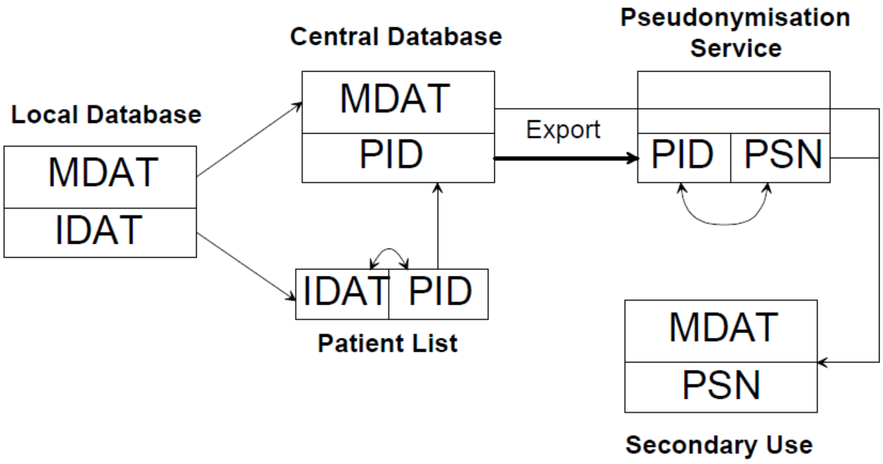


Figure 3.5: Pommerening: Data Flow for many Secondary Uses [35]

party institution. An attacker could bribe an insider of the third party institution to get access to the patient list or the identifying data of some pseudonyms.

3.5 Approach of Peterson

The System invented by Peterson [33] provides immediate access to personal medical information records, for example, in an emergency situation. Firstly, the patient registers an account via the Internet. Therefore, she has to enter a system wide unique private encryption key PEK and a password. Secondly, the server returns a unique global key GK, which has to be different from PEK. The global key GK could be printed on a card, which should be carried by the patient, for example in the wallet or as a necklace.

This approach consists of three database tables, the user table, the security table and the personal data table. The user table contains the GK, the PEK, a password and a foreign key to the security table. The security table contains a primary key, the method of encryption for the PEK, a server side encryption key and method and a foreign key to the personal medical data table. This table contains a primary key and the data, which is double encrypted with the PEK and the server side encryption key.

In case of an emergency, the health care personnel can retrieve the medical data of the patient by entering the global key GK, or if the patient is responsive to verbal commands, she can tell them the private encryption key PEK. The system looks up the database for the entered GK or PEK and returns the decrypted medical data. To modify or delete this medical data, the patient has to enter her password, which has been provided at registration time.

Table 3.3 shows different access levels of this approach. If a person knows the global key GK or PEK or both, but does not have a password, she is able to view medical data sets. To be able to add, modify or delete medical datasets, the person has to provide an additional password. Peterson argues, that these access levels protect the privacy of a patient, because the data does not contain any identifying information. So, for an attacker, it would be of no interest to receive anonymous data.

Global Key	*Personal Key*	*Password*	*Resultion Action*
No	No	No	Access Denied
Yes	No	No	View Only
No	Yes	No	View Only
Yes	Yes	No	View Only
No	No	Yes	Access Denied
Yes	No	Yes	View and Edit
No	Yes	Yes	View and Edit
Yes	Yes	Yes	View and Edit

Table 3.3: Approach of Peterson: Access Levels [33]

Fall-back mechanism

The approach of Peterson [33] provides a fall-back mechanism if the patient has lost her GK or wants to unauthorize her medical data. Therefore, the patient logs into the system with her PEK and password. Afterwards she requests a new GK which could be printed on a new card. The new GK assures, that her medical data is protected against unauthorized access if an old card GK is used.

Security issues

This approach has some major security issues. On the one hand, the data is stored doubly encrypted in the database. On the other hand the keys to decrypt these data are stored in the same database. So, if an attacker breaks into the database, she would be able to decrypt all data with the keys provided in the database. Furthermore, the password, PEK and GK of each user are stored in plain text. Therefore, an attacker may change the data stored in the database.

Another drawback is, that the user has to select the PEK manually and this key has to be unique. The system informs the user if the selected PEK is already in use. An attacker could use the keys, which are reported as already in use, to access all medical data which are associated with this key. Moreover, the technique to manually select a unique key through user input is unacceptable, impracticable and inefficient. For example, a new user may have to try out dozens of keys before a valid key is found.

3.6 Approach of Thielscher

Thielscher [58] proposed an electronic health record system, which uses decentralized keys stored on smart cards. The medical data is split into identification data and the anamnesis self and stored into two different databases. The key stored on the smart card of a patient is used to link the patient identity to her dataset. Therefore, this key generates a unique data identification code (DIC), which is also stored in the database. Such a DIC does not contain any information to identify an individual.

Data identification codes could be shared between the patient and health care providers to authorize them to be able to access the medical data set. For more security the authorization is limited to a time period. After this period any access attempts are invalid.

Furthermore, the system provides a mechanism in case of an emergency. Therefore some parts of the patient's individual health data is stored directly on the smart card. A health professional has immediate access to this data in case of an emergency. Moreover, the system includes an emergency call center which is authorized to access the central database for requests and to read the data in case of an emergency. Therefore, the health professional has to confirm their identity to the call center.

Fall-back mechanism

The keys to generate the data identification code (DIC) are stored on smart cards. In case these smart cards are lost, a fall-back mechanism is provided by Thielscher. Every

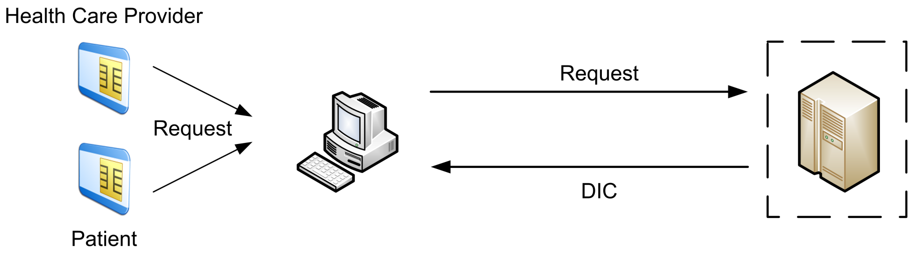


Figure 3.6: Thielscher: Architecture [58]

pseudonym held by a patient is stored on a list, which is saved on an off-line computer. In case the smart card is lost or destroyed, this list could be used to re-link the data to the patient.

Security issues

As already mentioned in the Approaches of Pommerening, a usage of a patients-pseudonyms list as a fall-back mechanism could lead to security issues. The work-around of Thielscher to keep the patients-pseudonyms list off-line promises a higher level of security, but does not prevent the system against a social-engineering attack on a system insider [2, 42]. Furthermore, this measure is useless if the attacker gains physical access to the computer.

Another drawback of the system is the emergency call center. This call center could abuse their access privileges to gain access to medical data of any patient.

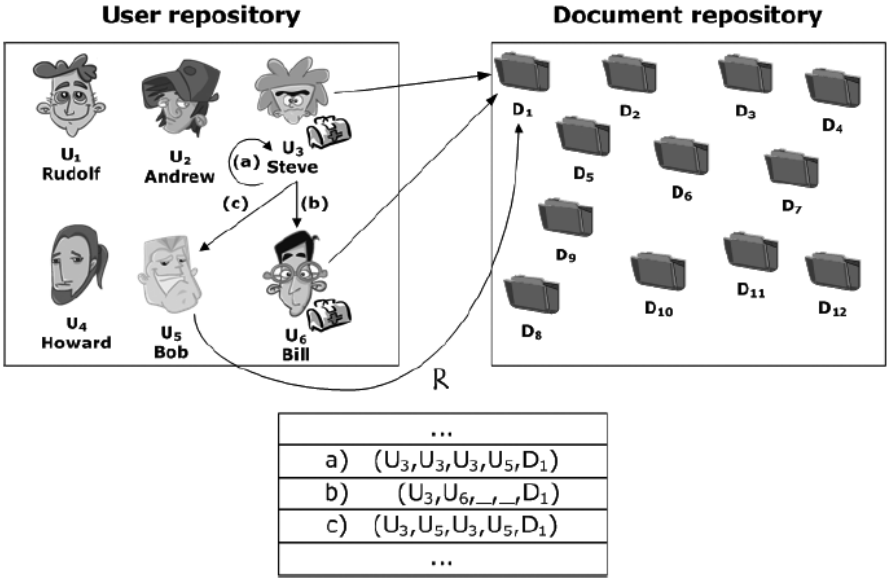
3.7 Approach of Slamanig and Stingl

The approach of Slamanig and Stingl [51, 53, 54] also uses pseudonymization to store the data in a centralized database and for authentication of smart cards but differs from PIPE and eGK. In contrast to PIPE and eGK, the system keeps the pseudonyms of a user secret. Each pseudonym represents a sub-identity of the user and is encrypted with its own public key. In case the user wants to view the datasets of one of her sub-identities, first she has to login to the system with her general pin code, and second, she has to enter the pin code of the sub-identity to activate the private key on the smart card. Furthermore, a public pseudonym of each user is available, which is used for authorization purposes.

Moreover, the system is divided into two repositories: the user repository and document repository. The link between these repositories is accomplished by holding a 5-tuple dataset $(U_S, U_R, U_C, U_P, D_i)$, which contains the sender U_S, the receiver U_R, the creator U_C, the concerning user U_P for example, the patient, and the document D_i. To ensure that during creation time no linkage between the concerned user is possible, all elements in the tuple are encrypted with the public key of the receiver, except of the receiver element U_R. Until the receiver has logged into the system, the receiver element U_R will be the public pseudonym. The next time the receiver logs in the system will replace the receiver element U_R with a secret pseudonym of the user and re-encrypt the other elements of the tuple.

As shown in figure 3.7, this tuple dataset can also be used for exchanging documents between users. There are six possible variations for exchanging or disclosing the medical documents [53].

1. $(\_,\_,U_C,U_P,D_i)$: Creator and concerning user are known
2. $(\_,\_,U_C,U_P*,D_i)$: Creator is known and concerning user is pseudonymized
3. $(\_,\_,U_C,\_,D_i)$: Creator is known and concerning user is anonymized
4. $(\_,\_,\_,U_P,D_i)$: Concerning user is known
5. $(\_,\_,\_,U_P*,D_i)$: Concerning user is pseudonymized
6. $(\_,\_,\_,\_,D_i)$: fully anonymized



	...
a)	(U3,U3,U3,U5,D1)
b)	(U3,U6,_,_,D1)
c)	(U3,U5,U3,U5,D1)
	...

Figure 3.7: Slamanig and Stingl: Repositories and Shares [51]

Fall-back mechanism

As a fall-back mechanism, the authors mentioned that a distributed key backup to N users using a (t, N)-threshold secret sharing scheme could be implemented, because the users private keys are essential for the system.

Security issues

The drawback of the approach of Slamanig and Stingl is, that an attacker may authorizes other users, send fraudulent medical documents or disclose medical data for a wrong person. This attack is possible, because the authors use a week mechanism for authorization or disclosure. For example, the requirements to send a fraudulent medical document are: (i) access to the database; (ii) the public pseudonym U_P of the user, which the attacker wants to harm; (iii) any public pseudonym to fake the sender U_S and creator U_C; (iv) the public pseudonym and the public key K_R of the receiver U_R, for example, the employer; and (v) a harmful document D_i. After the attacker has all the required information, she inserts a new tuple into the authorization table with the content found in equation 3.1. After the next login of the receiver, the system replaces the public pseudonym of the user with a private pseudonym of the receiver. The other attacks could be committed in the same way as demonstrated in the example.

$$(\{U_S\}_{K_R}, U_R, \{U_C\}_{K_R}, \{U_P\}_{K_R}, \{D_i\}_{K_R}) \tag{3.1}$$

CHAPTER 4

Evaluation and implications

This section compares the pseudonymization approaches, which have been presented above. Firstly, requirements are defined, which have to be met to be in accordance with the Data Protection Directive (95/46/EC) [12] in the European Union and the Health Insurance Portability and Accountability Act of 1996 (HIPAA) [59–61] in the United States.

4.1 Requirements for pseudonymization approaches

Pseudonymization systems have to adhere to the following requirements to be in accordance with privacy laws in the European Union or United States. The following set of requirements have been extracted from [12, 13, 17, 19, 20, 59–61]

- *User authentication*: The system has to provide adequate mechanisms for user authentication. This could be done, for example with smart cards or fingerprint readers.
- *Data ownership*: The owner of the data has to be the patient. She should be able to browse and read her medical records. Furthermore, the patient should be able to define who is authorized to access and create her medical records.
- *Limited access*: The system should provide data only to authenticated and authorized persons.

- *Protection against unauthorized and authorized access*: The records of an individual have to be protected against unauthorized access. Moreover, system administrators should also not be able to access these medical records, for example, through compromising the database.

- *Notice about uses of data*: The data owner should be informed about any access to her records.

- *Access and copy own data*: The system has to provide the possibility to patients to be able to access and copy their own data.

- *Disclosure option*: If records are disclosed, for example for secondary use, the data owner must have the possibility to decline or agree.

Additionally, a pseudonymization system should also implement the following functional requirements. These requirements are a subset of the requirements presented in [2, 39, 45].

- *Fall-back mechanism*: Systems should provide adequate possibilities to backup and restore the security token used for pseudonymization. Therefore, the system has to guarantee that the security token could only be restored under the compliance of a four-eye-principle.

- *Role-based access control*: is an approach to restricting functionality to authorized users with a specific role. For example, a patient should not be able to create a new dataset.

- *Unobservability*: means, that pseudonymized data could not be observed and linked to a specific individual in the system .

- *Secondary use*: The system should provide a mechanism to export pseudonymized data for secondary use and a possibility to notify the owner of the exported data if new medications or treatment methods are available.

- *Personal annotations*: The data owner should have the possibility to make annotations to the record.

- *Emergency access*: In case of an emergency, the rescue service or emergency physician should have access to an emergency dataset, in which important information is saved. For example, blood group, information about medication, allergic reactions to specific medicaments, etc.
- *Anonymized statistics*: The government needs reports about diagnoses or treatments to prevent epidemics and to control the health care system.

4.2 Accordance with legal and technical requirements

In table 4.1, the above defined requirements are compared with the laws and the pseudonymization approaches. The definition of the used symbols can be found in the legend table 4.2.

Requirements	*DPA*	*HIPAA*	*PIPE*	*eGK*	*Po*	*Pe*	*Th*	*Sl*
User authentication	x	x	x	x	-	o	x	x
Data ownership	x	x	x	x	-	-	x	x
Limited access	x	x	x	x	o	-	x	x
Protection against unauthorized and authorized access	x	x	x	x	o	-	o	x
Notice about uses of patients' data	x	x	x	x	-	-	-	-
Access and copy own data	x	x	x	x	o	x	x	x
Disclosure options	x	x	-	o	-	-	-	-

Additional requirements	*DPA*	*HIPAA*	*PIPE*	*eGK*	*Po*	*Pe*	*Th*	*Sl*
Fall-back mechanism	-	-	x	x	-	o	x	x
Role-based access control	-	-	x	x	-	-	x	x
Unobservability	x	x	x	x	x	-	x	x
Secondary use	-	x	x	o	x	-	-	x
Personal annotations	-	-	-	-	-	-	-	-
Emergency access	-	-	o	x	-	x	-	-
Anonymized statistics	-	x	o	-	o	-	-	x

Table 4.1: Comparison of pseudonymization approaches in accordance with the laws [26–28]

Table 4.1 shows that newer approaches implements more features to in accordance with the laws in the European Union and the United States. Most of the approaches implement

Abbreviations		
Po	...	approach of Pommerening
Pe	...	approach of Peterson
Th	...	approach of Thielscher
Sl	...	approach of Slamanig and Stingl

Legend for DPA and HIPAA		
x	...	defined and accurate with the law
-	...	undefined in the law

Legend for pseudonymization approaches		
x	...	fully implemented
o	...	partially implemented
-	...	not implemented

Table 4.2: Legend for table 4.1

the requirements of *user authentication*, *data ownership*, *limited access* and provide control mechanisms *against unauthorized and authorized access.* The implementation of the requirements *notice about uses of patients data* and *disclosure options* is inadequate. The data owner needs the possibility to give consent to the disclosure of her pseudonymized data by law.

Moreover, table 4.1 shows that additional requirements, which enhance the security of the system and the containing datasets, are widely implemented. None of the above mentioned approaches implement a possibility for personal annotations.

Common requirements such as *emergency access*, *electronic prescriptions* and *anonymized statistics* are currently under development and implemented in some approaches.

4.3 Possible security issues of pseudonymization approaches

This section presents a list of possible security issues and compares it with the pseudonymization approaches mentioned above.

- *Insider abuse*: Medical personnel may abuse their access rights for their own purposes. For example, they may want to know how family members or celebrities are being treated [42]. Insiders do not only abuse their privileges for their own purposes, they may release information to outsiders for spite, revenge or profit [42].

- *Social engineering*: is a common method to get information about a person. Therefore, an attacker could bribe or mislead an administrator of the pseudonymization system. For example, the attacker could fake her identity to get a new security token.
- *Disclosure of weakly pseudonymized data*: Data, which is only pseudonymized by removing the name of the patient, could lead to a data mining attack as shown by Sweeny in [57]. Sweeny was able to combine medical data with an electronic version of a city's voter list.
- *Attacker steals database*: An attacker, who could access and copy the electronic health record database, could possibly commit a data profiling attack. Thus, the attacker can collect statistics and information about the data. In the worst case scenario the attacker could reconstruct the pseudonyms.
- *Attacker deletes data*: If an attacker breaks into the system, she may have the possibility to delete data. Therefore the system should be able to detect such changes and inform the system administrator about this attack and request a restoration of the datasets.
- *Attacker modifies data*: An attacker, who has broken into the system, may also change some datasets. Therefore, the system should digitally sign all records to be able to detect such a modification.
- *Attacker authorizes other users*: An attacker could try to authorize another user or herself to be able to gain access to medical data of other users.
- *Administrator accesses data*: Administrators of the pseudonymization system could access the completely linked database if the data is pseudonymized by disclosure only.
- *Administrator accesses cryptographic keys*: If system administrators have access to the private keys of individuals, she may have the possibility to decrypt all pseudonyms and link anamnesis to individuals. This case could lead to insider abuse, which is described above. Moreover, every attacker who has administration privileges could

steal the database with all the keys.

Possible security issues	*PIPE*	*eGK*	*Po*	*Pe*	*Th*	*Sl*
Insider abuse	-	-	x	x	x	-
Social engineering	o	o	x	x	x	o
Disclosure of weakly pseudonymized data	x	-	x	x	x	x
Attacker steals database	o	o	o	x	o	o
Attacker deletes data	x	x	x	x	x	x
Attacker modifies data	x	-	x	x	x	x
Attacker authorizes other users	-	-	-	-	-	x
Administrator accesses data	-	-	x	x	-	-
Administrator accesses cryptographic keys	-	-	o	x	o	-

Table 4.3: Comparison of security issues and pseudonymization approaches

Abbreviations		
Po	...	approach of Pommerening
Pe	...	approach of Peterson
Th	...	approach of Thielscher
Sl	...	approach of Slamanig and Stingl

Legend for pseudonymization approaches		
x	...	security issue
o	...	possible security issue
-	...	no security issue

Table 4.4: Legend for table 4.3

Table 4.3 shows that current implementations of pseudonymization approaches could not prevent all variations of possible attacks. However, we have to distinguish between pseudonymization systems which only pseudonymize data on export, for example for secondary use, and pseudonymization systems which store the data as pseudonymized in the database.

The approaches of Pommerening and Peterson only pseudonymize data on export. These systems permit attackers to steal the database with all data linked to individuals. Moreover, system administrators could abuse their access privileges to release information to outsiders for revenge, profit or their own purposes [42].

In his approach, Peterson tried to prevent the following types of attacks: insider abuse, disclosure of weakly pseudonymized data, and databases being stolen. He did so by defining that no identifiable data is allowed to be stored. However, the system is not able to check if identifiable words exist in the data.

The weak point of Thielscher's approach is that a centralized pseudonym list exists to unlock patient mapping for recovery purposes. To prevent attacks to the list, Thielscher keeps it off-line, but that measure could not prevent insider abuse or social engineering attacks. Furthermore, the emergency call center has full access to all medical data.

In contrast, PIPE, eGK and the approach of Slamanig and Stingl are pseudonymization systems which store the data as pseudonymized in the database. Attackers who have stolen the database or system administrators would not be able to link the data to individuals. One way to link the data to an individual would be to commit a social engineering attack, another way would be to fake the identity of the person whom the attacker wants to attack. Additionally, this attacker would have to fake an official photo identification, like a passport or driver's license, to get a new smart card to access the system. Another method to link data to an individual would be to carry out a data mining or data profiling attack. These attacks could be done by using identified keywords, which are not encrypted in the eGK approach or by using identifiable words in the anamnesis data in the PIPE approach or in the approach of Slamanig and Stingl.

CHAPTER 5

Core workflows of PIPE

In this chapter, core workflows such as add an actor, add data, search data, retrieve data and recover a lost card of PIPE will be presented. These workflows have been created after the evaluation of the first prototype of PIPE and implemented into a second prototype.

The first prototype has been developed in Java and does not use smart cards. The smart cards have been simulated using a database table and java cryptographic functions. This prototype has been developed to prove theoretical propositions and for evaluating the performance of PIPE.

The results of the first prototype have been evaluated and implemented into a second prototype. This prototype has been developed in *C# .net* to ensure the support of smart cards. To simplify the implementation, this prototype does not use a client-server architecture as that will be the case in the final version of PIPE. But the interfaces of API, ADMIN and SERVICE would be the same.

5.1 Add an actor to the system

This workflow describes how to add a new actor to the system. In this workflow all necessary cryptographic keys and the associated smart card are created. Moreover, the inner private key and inner symmetric key are encrypted and stored in the database.

Furthermore, the external user ID of the legacy application is mapped to the new internal user.

First of all, the ADMIN requests the SERVICE for a new unique internal user ID. This internal user ID, a generated asymmetric key pair and a pin code to protect these keys, are stored on a new smart card. To be able to restore the inner private key in case the smart card is lost, stolen or just wears out, the system shares this inner private key with N_a operators of the system. Therefore, *(5)* the inner private key and *(6)* the internal user ID are encrypted with the public key of the logic (SERVICE) and sent over the network to the SERVICE. There, *(8)* the inner private key and *(9)* the internal user ID are decrypted by the logic (SERVICE). Next, *(10)* the internal user ID is encrypted with the symmetric key of the logic (SERVICE) and *(11)* N_a shares are generated, for example, with the Shamir threshold scheme [49]. These N_a shares are *(12)* encrypted with the symmetric key of the logic (SERVICE) and sent with the encrypted internal user ID to a randomly selected operator. There, *(14)* the share and the internal user ID are encrypted with the symmetric key of the operator and inserted into the database.

Next, if all shares have been successfully shared, *(18)* the inner symmetric key is generated and *(20)* encrypted with the inner public key. Furthermore, *(19)* the inner private key is encrypted with the outer public key of the smart card. Finally, *(21)* the keys are stored in the database.

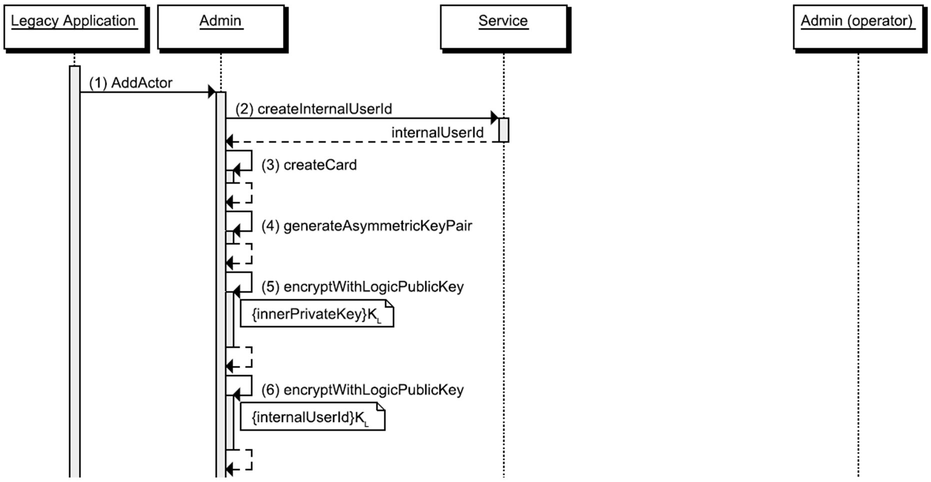


Figure 5.1: Sequence-Diagram: Add an Actor to the system (part 1)

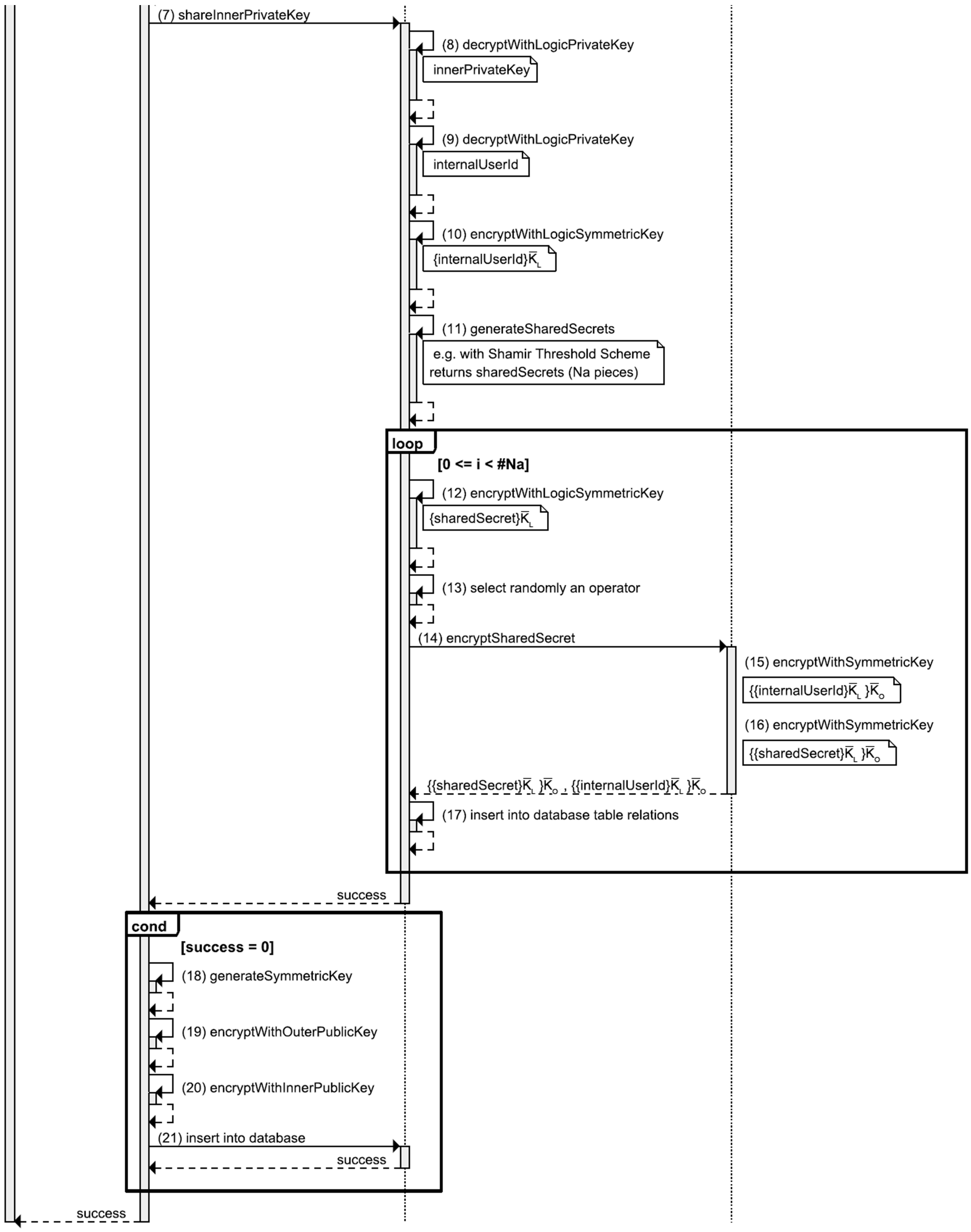

(7) shareInnerPrivateKey
(8) decryptWithLogicPrivateKey
innerPrivateKey
(9) decryptWithLogicPrivateKey
internalUserId
(10) encryptWithLogicSymmetricKey
{internalUserId}$\overline{K}_L$
(11) generateSharedSecrets
e.g. with Shamir Threshold Scheme
returns sharedSecrets (Na pieces)
loop
[0 <= i < #Na]
(12) encryptWithLogicSymmetricKey
{sharedSecret}$\overline{K}_L$
(13) select randomly an operator
(14) encryptSharedSecret
(15) encryptWithSymmetricKey
{{internalUserId}$\overline{K}_L$ }$\overline{K}_o$
(16) encryptWithSymmetricKey
{{sharedSecret}$\overline{K}_L$ }$\overline{K}_o$
{{sharedSecret}$\overline{K}_L$ }$\overline{K}_o$, {{internalUserId}$\overline{K}_L$ }$\overline{K}_o$
(17) insert into database table relations
success
cond
[success = 0]
(18) generateSymmetricKey
(19) encryptWithOuterPublicKey
(20) encryptWithInnerPublicKey
(21) insert into database
success
success

Figure 5.2: Sequence-Diagram: Add an Actor to the system (part 2)

5.2 Add data to the system

This workflow describes how to add new data to the system. The data will be stored completely pseudonymized. The data could either be a clinical report, some information which has been entered by a doctor or a foreign key of another database table. The data will be assigned to the concerning user, for example, the patient. Moreover, this user can manage the access rights of other users, for example, health care providers, for this data.

Firstly, the user authenticates against the system. Therefore, the function *(2) authenticateUser* unlocks the keys on a smart card with the given pin code. Next, the API encrypts *(3)* the internal user ID and *(4)* the selected keywords with the users inner symmetric key. Furthermore, the SERVICE creates and sends back *(5) - (8)* a new unique pseudonym, which is encrypted with the public key of the user. The API decrypts *(9)* the pseudonym and encrypts *(10)* it with the inner symmetric key of the user. Finally, *(11)* the encrypted parameters, such as internal user ID, keywords and pseudonym are sent to the SERVICE and stored in the database. If the operation has not been successfully executed, the SERVICE and API return an error code.

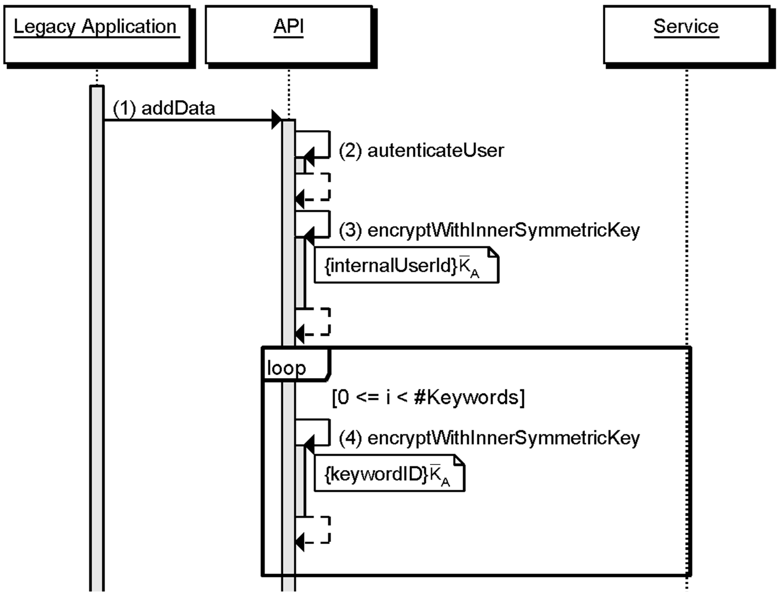


Figure 5.3: Sequence-Diagram: Add data to the system (part 1)

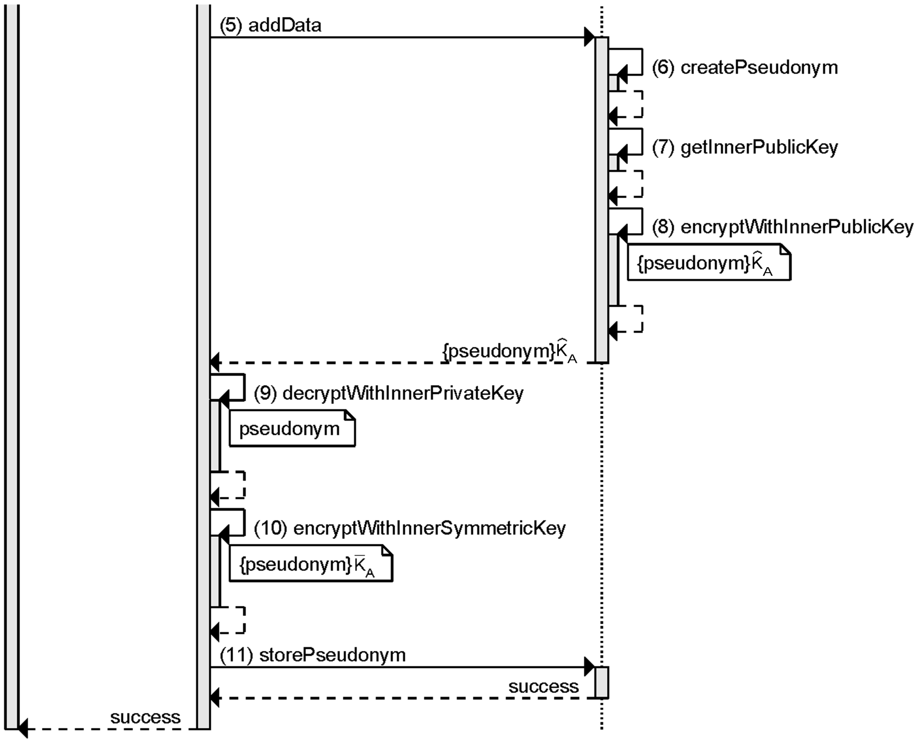

(5) addData
(6) createPseudonym
(7) getInnerPublicKey
(8) encryptWithInnerPublicKey
{pseudonym}$\widehat{K}_A$
{pseudonym}$\widehat{K}_A$
(9) decryptWithInnerPrivateKey
pseudonym
(10) encryptWithInnerSymmetricKey
{pseudonym}$\overline{K}_A$
(11) storePseudonym
success
success

Figure 5.4: Sequence-Diagram: Add data to the system (part 2)

5.3 Search data in the system

This workflow describes how to search for stored and pseudonymized data. This method could be used as an infunction of other functions, for example, the *retrieveData* method needs a pseudonym, which has been searched for first.

First of all, the user is authenticated against the system with the *(2) authenticateUser* method. Next, the API encrypts *(3)* the internal user ID and *(4)* all selected keywords. Next, *(5)* the encrypted parameters are sent to the SERVICE, and the SERVICE selects all pseudonyms which match the encrypted internal user ID and the keywords and then returns the list of encrypted pseudonyms. Finally, *(6)* the API decrypts the pseudonyms and returns the list of valid decrypted pseudonyms to the legacy application. Due to the decryption of the pseudonyms, the system can verify that the pseudonyms belong to the logged in user.

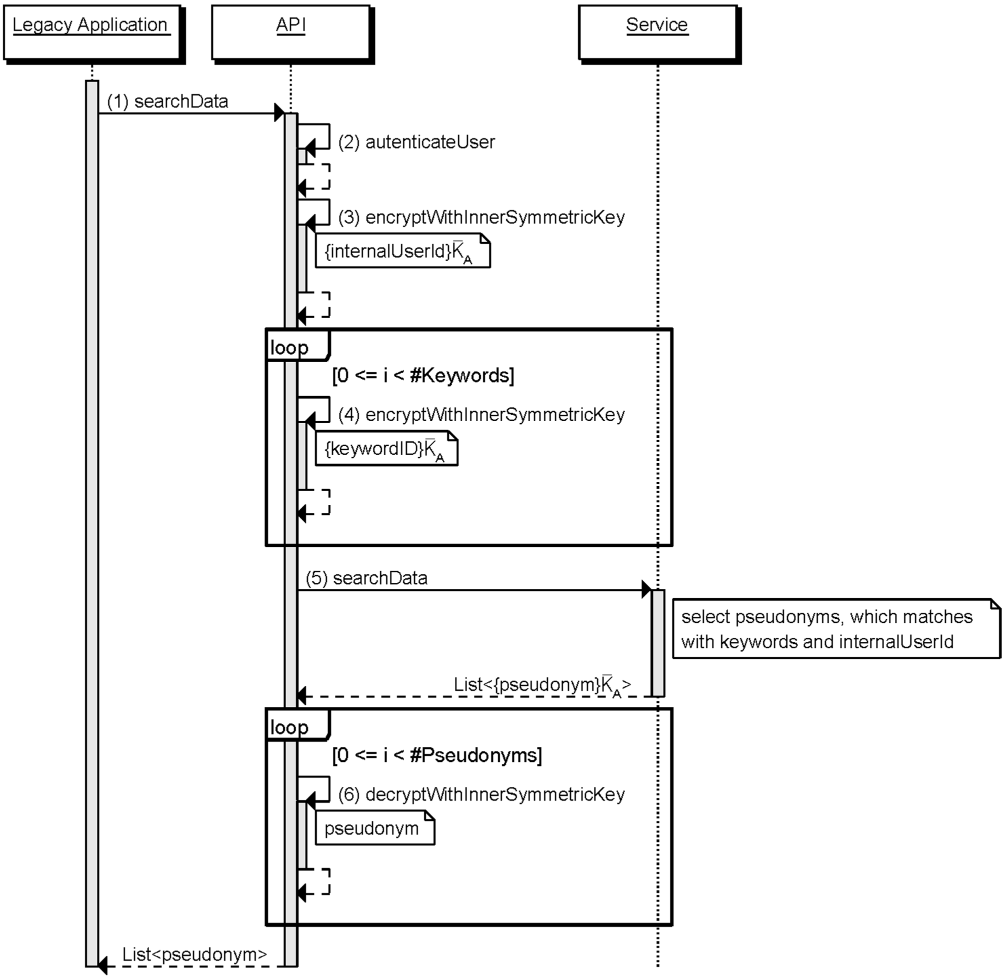

Legacy Application
API
Service
(1) searchData
(2) autenticateUser
(3) encryptWithInnerSymmetricKey
{internalUserId}$\bar{K}_A$
loop
[0 <= i < #Keywords]
(4) encryptWithInnerSymmetricKey
{keywordID}$\bar{K}_A$
(5) searchData
select pseudonyms, which matches with keywords and internalUserId
List<{pseudonym}$\bar{K}_A$>
loop
[0 <= i < #Pseudonyms]
(6) decryptWithInnerSymmetricKey
pseudonym
List<pseudonym>

Figure 5.5: Sequence-Diagram: Search data in the system

5.4 Retrieve data from the system

This workflow describes how to retrieve stored and pseudonymized data from the PIPE.

To retrieve stored data from PIPE, a decrypted pseudonym is needed, which has been returned by the *serachData-Method* first. After the API has authenticated the user against the system, the pseudonym is sent to the SERVICE. Next, the SERVICE selects the data which belongs to the pseudonym from the database and sends it back to the API, where it passes-through the data to the legacy application.

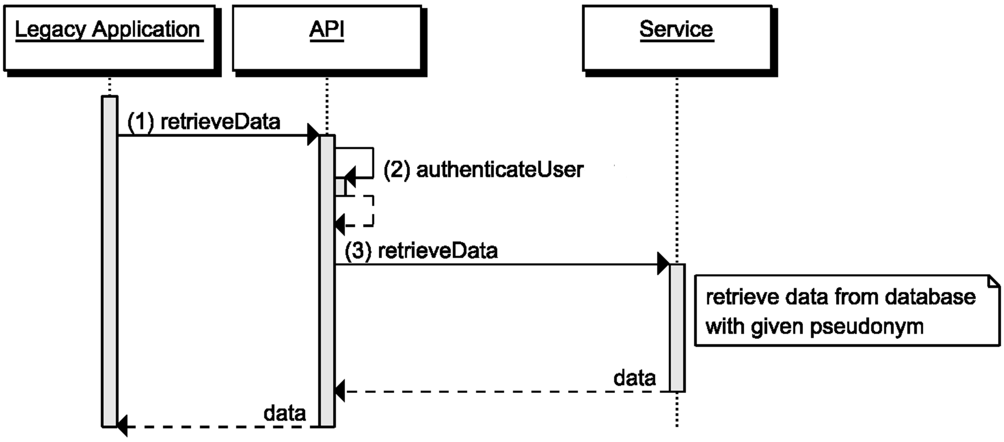


Figure 5.6: Sequence-Diagram: Retrieve data from the system

5.5 Recover lost card

This workflow describes how to produce a new card with new keys if the current card is lost, stolen or just worn out.

Firstly, the ADMIN encrypts *(2)* the internal user ID of the concerning user with the public key of the logic (SERVICE) and sends it to the SERVICE. There, the internal user ID will be *(4)* decrypted and *(5)* encrypted with the symmetric key of the SERVICE. Next, the SERVICE broadcasts *(6)* the encrypted internal user ID to all currently available operators. Each operator encrypts the encrypted *(7)* internal user ID a second time with the operator's inner symmetric key and *(8)* selects the shared piece of the users inner private key and if available, sends it back to the SERVICE.

If the SERVICE has retrieved N_k shared pieces, *(11)* the key will be reproduced with the use of Shamir's threshold scheme [49]. The reproduced inner private key is encrypted *(12)* with the public key of the operator who submitted the request and sent back to her. There, *(14)* the new card is created and *(15)* the inner public key is encrypted with the new outer public key. After the new card has been created, *(17)* the inner private key is updated in the database, and *(18)* the old shares are deleted, and the inner private key is shared again with other operators.

After this workflow, the old smart card is useless, because the inner private key is encrypted with a new outer public key, which is only available on the new smart card.

Moreover, to enhance the security of this process, the last steps beginning from step *(12)* to *(18)* could also be done on the server-side in the SERVICE. This security extension will prevent the unencrypted inner private key of the user who lost their card from being available unencrypted in the memory of the operator computer. Furthermore, the misuse of this process by the operator could be prevented, because the newly generated smart card could be sent by mail.

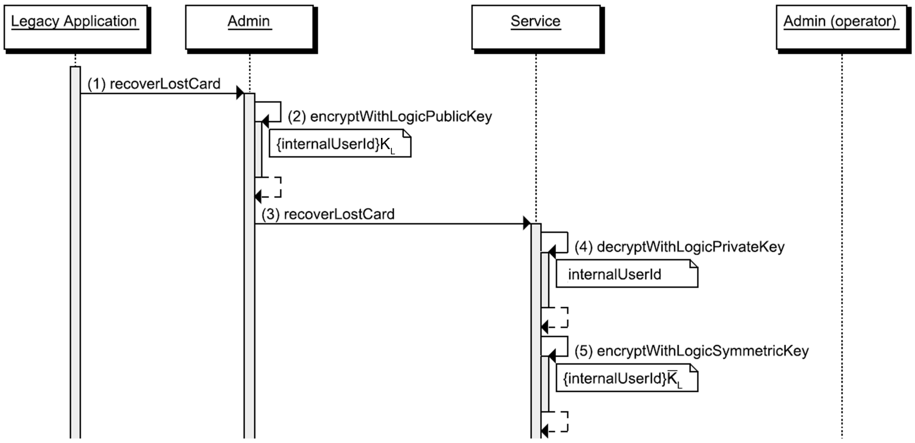

Legacy Application
Admin
Service
Admin (operator)
(1) recoverLostCard
(2) encryptWithLogicPublicKey
{internalUserId}K_L
(3) recoverLostCard
(4) decryptWithLogicPrivateKey
internalUserId
(5) encryptWithLogicSymmetricKey
{internalUserId}$\overline{K}_L$

Figure 5.7: Sequence-Diagram: Recover lost card (part 1)

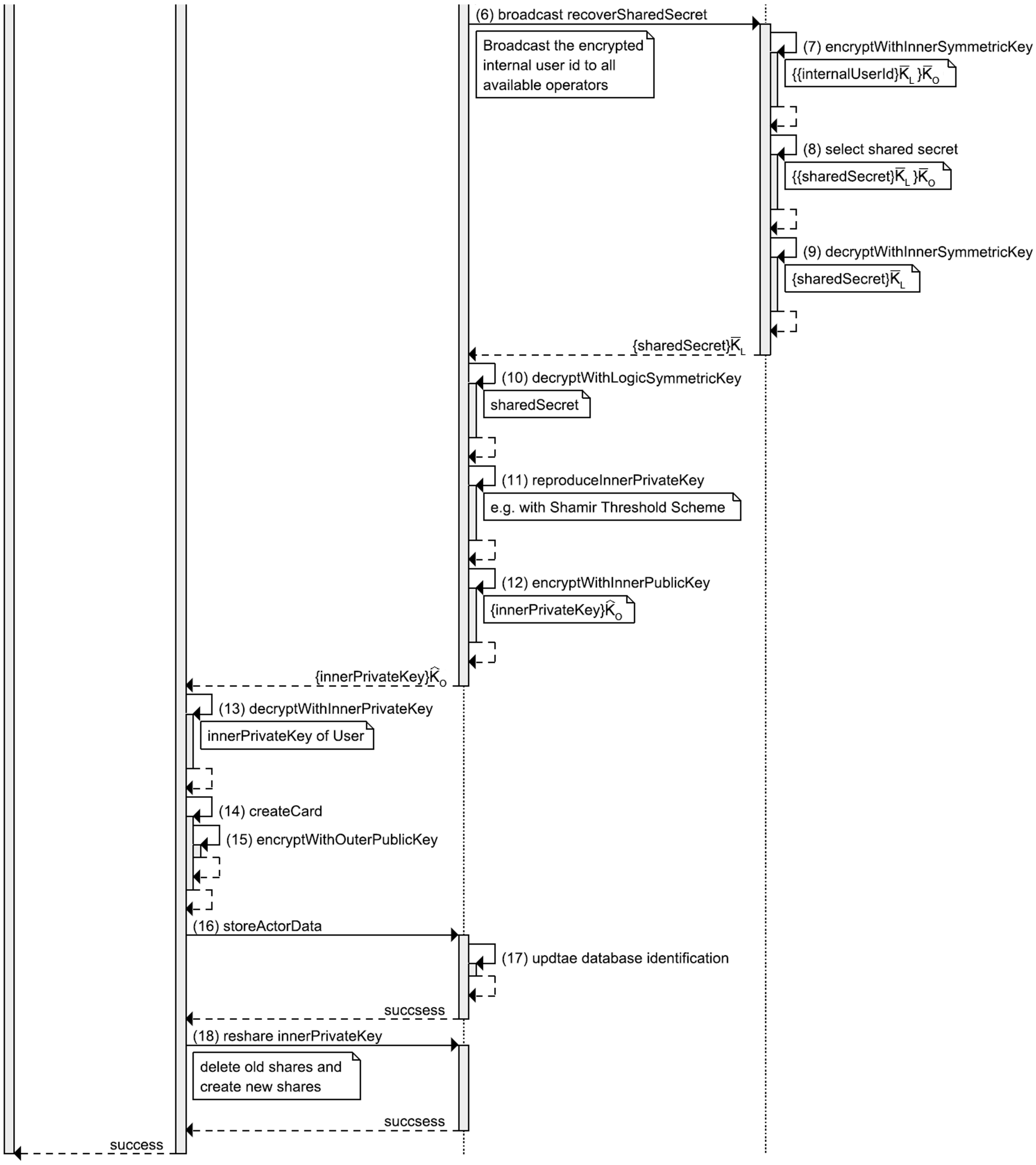

(6) broadcast recoverSharedSecret
Broadcast the encrypted internal user id to all available operators
(7) encryptWithInnerSymmetricKey
{{internalUserId}$\overline{K}_L$ }$\overline{K}_O$
(8) select shared secret
{{sharedSecret}$\overline{K}_L$ }$\overline{K}_O$
(9) decryptWithInnerSymmetricKey
{sharedSecret}$\overline{K}_L$
{sharedSecret}$\overline{K}_L$
(10) decryptWithLogicSymmetricKey
sharedSecret
(11) reproduceInnerPrivateKey
e.g. with Shamir Threshold Scheme
(12) encryptWithInnerPublicKey
{innerPrivateKey}$\overline{K}_O$
{innerPrivateKey}$\widehat{K}_O$
(13) decryptWithInnerPrivateKey
innerPrivateKey of User
(14) createCard
(15) encryptWithOuterPublicKey
(16) storeActorData
(17) updtae database identification
succsess
(18) reshare innerPrivateKey
delete old shares and create new shares
succsess
success

Figure 5.8: Sequence-Diagram: Recover lost card (part 2)

CHAPTER 6

Architectural design of PIPE's fall-back mechanism with HSM

6.1 Hardware Security Module - Introduction

6.1.1 Installing nShield Hardware Security Module

1. On Windows Operating Systems, the driver has to be installed before the PCI module is plugged in (Hardware Installation Guide).
2. Testing the installation (nShield Administrator Guide, page 46)
    a) Log in as normal user
    b) Run *enquiry.exe* in command line
    c) If the server service (hardserver.exe) is not running, restart it.
3. Testing the Smart Card Reader
   slotinfo.exe -m MODULE [-s SLOT] (In our case: -m 1 -s 0)

   Required output

   ```
   Module n slot 0:
   Token not formatted
   ```

   or

```
Module n slot 0:
Authentication key: 00000000-00000000-00000000-00000000-00000000
No data on token
3698 bytes free
```

6.1.2 Configure nShield Hardware Security Module

1. Login as user with administration rights.
2. The configuration file is located at: c:\kmdata\config
3. Stop the nfast server service.
4. Configure config as described in the file, additional help can be found in nShield Administrator Guide, page 85. server port: 9004
   non privileged connection port: 9000
   privileged connection: 9001
5. Start the nfast service.
6. Run the cfg-reread command-line utility to load the new configuration.
7. To test if the hardserver is configured correctly, use *enquire.exe*.
8. Run *nfkminfo.exe* to test that the client has access to the data of the security world. Required state *0x2 Usable*

6.1.3 Creating and configuring a security world

1. To create a security world the HSM has to be in pre-initialization state. (change hardware button from O to I)
2. The logged on user has to be permitted to create privileged connections.
3. The NFAST_HOME environment variable must be set. (should be done by the installation program)

4. To create a new world, use the *new-world.exe* command line tool (nShield Administrator Guide, page 57)

   ```
   new-world [-i|--initialize] [-S|--no-remoteshare-cert]
   [-o|--overwrite] [-F|--strictfips-140-2-level-3]
   [-R|--no-recovery] [-tTIMEOUT|--nso-timeout=TIMEOUT]
   [-m|--module=MODULE] [-Q|--acs-quorum=K/N] [FEATURES]
   ```

5. A security world could be created with the *keysafe.exe* too. (nShield Administrator Guide, page 131)
6. Switch the HSM to operational mode. (change hardware button from I to O)

6.1.4 How to use the nShield Hardware Security Module with Java

The nShield hardware security module can be accessed through the Java JCA/JCE architecture. Therefore, nChipher has included a JCA/JCE provider. First of all, this JCA/JCE provider has to be added to the java runtime environment. This can be done statically by editing the file located at *%JAVA_HOME%/lib/security/java.security* as shown in listing 6.1 or dynamically loaded at runtime as shown in listing 6.2 line 9, and the following *JAR-Files* have to be included in the classpath: *jcetools.jar, jutils.jar, keysafe.jar kmjava.jar, kmcsp.jar, nfjava.jar, rsaprivenc.jar, spp.jar.*

```
...
#
# List of providers and their preference orders:
#

# Add nCipher provider (as default provider)
security.provider.1=com.ncipher.provider.km.nCipherKM

# Other provider
security.provider.2=sun.security.provider.Sun
...
```

Listing 6.1: Add nCipher provider (static method)

```
1  import java.io.*;
   import java.security.*;
3  import javax.crypto.*;

5  public class SimpleExample {

7      public static void main(String[] args) {

9        Security.addProvider(new com.ncipher.provider.km.nCipherKM());
```

Listing 6.2: Add nCipher provider (dynamic method)

In listings 6.2, 6.3, 6.4 and 6.5, an example will be given how to use the nCipher JCA/JCE provider. In this example, there is no any exception handling implemented to keep it as simple as possible. After the security provider has been loaded and a HSM operator card is inserted into the HSM's card reader slot, all features of the provider can be used.

In listing 6.3, a request for a reference to the *Security World* of the nShield HSM is shown on line 16. The name of the *Security World* is *nCipher.sworld* and is available in the nCipher provider *nCipherKM*. On line 17, the *KeyStore* is loaded, which is located in the *Security World* of the hardware security module. A *KeyStore* can be used to manage keys and certificates [55]. The file *keystore* contains only a HSM internal reference number, which points to the requested *KeyStore*.

In listing 6.4, is demonstrated how to create a new key or load an existing key. First of all, the creation of a new Rijndael (AES) key is described. Therefore, a reference to the *KeyGenerator* for the Rijndael algorithm is requested, which is available in the nCipher provider *nCipherKM* as shown on line 25. Furthermore, the *KeyGenerator* is initialized to create a 128bit AES key and to create the key in the hardware security module as shown

```
10       Key key;
         String keyname, text, mode;
12       BufferedReader in = new BufferedReader(new InputStreamReader(System.in));

14       try {

16         KeyStore ks = KeyStore.getInstance("nCipher.sworld", "nCipherKM");
           ks.load(new FileInputStream("keystore"), null);
```

Listing 6.3: How to use the JCE/JCA provider: load KeyStore

on line 27. The returned *key* object is a proxy, which points to the key in the HSM to avoid having the key in the local memory.

To add the key to the *KeyStore* a unique alias name has to be chosen. The third parameter of the *setKeyEntry* method specifies a password for the key. In the example, the password is *null*, because the key is protected with an operator card set where no password has been set. In case the key is a private key, the fourth parameter is used to certify the corresponding public key.

Finally, the *KeyStore* is saved in the *keystore* file. In this file the nCipher provider saves only an internal reference number to identify the *KeyStore* in the hardware security model.

Second of all, how to retrieve an existing key from the hardware security module is described. Therefore, the *KeyStore* has to be loaded successfully. Next, the *getKey* method with the alias name of the key and if necessary, a password, is called as shown on line 31. As already mentioned, the returned *key* object is a proxy, which points to the key in the hardware security module.

In listing 6.5, the cipher algorithm is loaded and the encryption or decryption of data is done. Therefore, the *cipher* object to the Rijndael algorithm available in the nChipher provider *nCipherKM* is set as shown on line 33. Next, the *cipher* object is initialized for encryption or decryption as shown on line 44 and 46. Finally, the encryption or decryption

```
18          System.out.println("Do you want to create a new key? [Y|N]");
            mode = in.readLine();
20
            System.out.println("Enter a name for the key:");
22          keyname = in.readLine();

24          if(mode.equalsIgnoreCase("Y")) {
              KeyGenerator kg = KeyGenerator.getInstance("Rijndael", "nCipherKM");
26            kg.init(128);
              key = kg.generateKey();
28            ks.setKeyEntry(keyname, key, null, null);
              ks.store(new FileOutputStream("keystore"), null);
30          } else {
              key = ks.getKey(keyname, null);
32          }
```

Listing 6.4: How to use the JCE/JCA provider: create a new key or load a existing key

is done by calling the *doFinal* method of the *cipher* object. The data, which should be encrypted or decrypted, is passed as a parameter of this method, as shown on line 49.

```
        Cipher cipher = Cipher.getInstance("Rijndael", "nCipherKM");
34
        System.out.println("Enter Text: ");
36      text = in.readLine();

38      byte[] data = text.getBytes();

40      System.out.println("encrypt or decrypt? [E|D]: ");
        mode = in.readLine();
42
        if(mode.equalsIgnoreCase("E")) {
44        cipher.init(Cipher.ENCRYPT_MODE, key);
        } else {
46        cipher.init(Cipher.DECRYPT_MODE, key);
        }
48
        byte[] result = cipher.doFinal(data);
50
        System.out.println("Original data : " + new String(text));
52      System.out.println("Encrypted or decrypted data: " + new String(result));

54    } catch (Exception e) { }
     }
56  }
```

Listing 6.5: How to use the JCE/JCA provider: encrypt/decrypt data

Moreover, the nShield Hardware Security Module can be used as a cryptographic accelerator device. Therefore, the encryption or decryption functions allow the usage of external cryptographic keys.

In listing 6.6 a short example is shown, which introduces how to use external keys with the nShield HSM. First a symmetric key is created with the standard SunJCE cryptographic provider, which does not use any parts of the nShield HSM, on line 15. Next, the Rijndael chipher object is loaded from the nChipher provider on line 20. On the next line, this cipher is initialized with the external symmetric key, which has been created above. On lines 23-30, a text is encrypted and decrypted with the Rijndeal cipher from the nChipher provider.

```java
import java.security.Key;
import java.security.Security;
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;

public class ExternalSymKey {
  public static void main(String[] args) {

    Security.addProvider(new com.ncipher.provider.km.nCipherKM());
    Key key;

    try{
      KeyGenerator kg = KeyGenerator.getInstance("AES", "SunJCE");
      kg.init(256);
      key = kg.generateKey();

      byte[] data = "Hello World!".getBytes();
      System.out.println("Original data : " + new String(data));

      Cipher cipher = Cipher.getInstance("Rijndael", "nCipherKM");

      cipher.init(Cipher.ENCRYPT_MODE, key);
      byte[] result = cipher.doFinal(data);

      System.out.println("Encrypted data: " + new String(result));

      cipher.init(Cipher.DECRYPT_MODE, key);
      byte[] original = cipher.doFinal(result);

      System.out.println("Decrypted data: " + new String(original));
    } catch (Exception e) { }
  }
}
```

Listing 6.6: How to use the JCE/JCA provider: encrypt/decrypt data with external keys

6.2 PIPE's fall-back mechanism with HSMs

In this section, the fall-back mechanism of PIPE, which is described in section 3.2, is extended to support hardware security modules as additional operators O. The goal of the architecture is to gain the optimal trade-off between security on the one hand and usability, performance and cost on the other hand. To prevent the system from gaining access to critical security parameters, a HSM with FIPS Level 3 [15] enabled is recommended. The tests in this chapter have been performed on an nCipher nShield PCI 500.

In contrast to the fall-back mechanism without HSM, the extended version with HSMs enhances the protection against misuse of the rights, which have been given to the operators of the system. In table 6.1, the definition of the additional system attributes for PIPE with HSMs can be found.

	Operator	*Human O*	*Machine O*	*Logic*
abbreviation	O	H	M	L
unique identifier	O_{id}	H_{id}	M_{id}	
(outer public key, private key)	(K_O, K_O^{-1})	(K_H, K_H^{-1})	(K_M, K_M^{-1})	
(inner public key, private key)	$(\widehat{K}_O, \widehat{K}_O^{-1})$	$(\widehat{K}_H, \widehat{K}_H^{-1})$	$(\widehat{K}_M, \widehat{K}_M^{-1})$	$(\widehat{K}_L, \widehat{K}_L^{-1})$
inner symmetric key	$\overline{K}_O$	$\overline{K}_H$	$\overline{K}_M$	K_L
key share	$\sigma_\iota(K)$	$\sigma_{\mathcal{H}_\iota}(K)$	$\sigma_{\mathcal{M}_\iota}(K)$	
medical data / anamnesis	φ_i			
pseudonym	ψ_{i_j}			

Table 6.1: PIPE: Definition of additional system attributes with HSM

First of all, a two-step variant of Shamir's threshold scheme [49] will be applied. This threshold scheme can be defined for sharing the user's inner private key, (i) the number of shares n and (ii) the amount of shares that are necessary to re-establish the shared key. Hence, as higher the number of issued shares compared to the number of shares in total, as higher the security, if the operators have been selected randomly and each of them are holding exactly one share.

To enhance the security of the fall-back mechanism, the user's inner private key $\widehat{K_A^{-1}}$ is shared with human operators H and machine operators M. Therefore the inner private key is divided into two parts, $\sigma_{\mathcal{H}}(\widehat{K_A^{-1}})$ for the human operators and $\sigma_{\mathcal{M}}(\widehat{K_A^{-1}})$ for the machine operators as shown in figure 6.1. Next, the threshold scheme is applied to each resulting part. These parts are subdivided into the number of assigned human and machine operators. Finally, PIPE distributes the shares encrypted with the system's symmetric key to the randomly selected operator. There, the operator O encrypts the share with her symmetric key $\overline{K}_O$, which has also been done in the version of PIPE without a hardware security module.

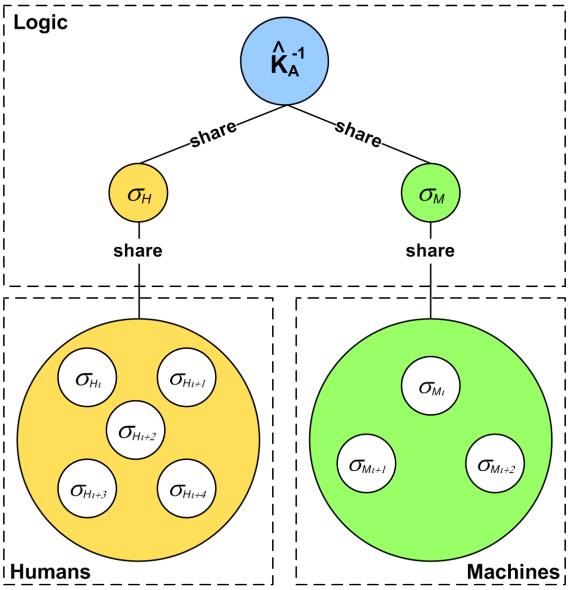


Figure 6.1: PIPE: Two-folded variant of Shamir's threshold scheme [41]

6.3 Security investigation of the fall-back mechanism with HSM

This section investigates the security of the PIPE's fall-back mechanism with the use of hardware security modules. Following Shamir [49], it is not possible to combine fewer shares than needed $(k - 1)$ to compute the user's inner private key. Otherwise, if an attacker is able to bribe more than or equal of needed operators $(b \geq k)$, she may be able to reconstruct a certain user's identity, but therefore the attacker has to acquire the system's symmetric key.

To illustrate the security of PIPE's fall-back mechanism, an example will be given in the next paragraph. Under the assumption that the operators do not know whose shares they are holding, the probability for determining the least number of operators for a specific user is calculated with the equation shown in 6.1. Furthermore, a successful attack is defined by bribing at least $b_H \geq k_H$ human operators and $b_M \geq k_M$ machine operators. Moreover, bribing $b_H \geq k_H$ human operators also does not influence the probability of recovering the machine operators' sub secret $\sigma_{\mathcal{H}}$. Therefore, these two events can be defined as statistically independent. Following the multiplication rule for independent events, the probability for their intersection, which means combining all necessary shares of a specific secret, is equivalent to the product of the single probabilities [41].

$$P(k \leq X \leq n) = \sum_{\iota=k}^{n} \frac{\binom{n}{\iota}\binom{|\mathcal{O}|-n}{b-\iota}}{\binom{|\mathcal{O}|}{b}} \tag{6.1}$$

The result of equation 6.1 applied to an example is shown in figure 6.2. The configuration of the figure is $k_{\mathcal{H}} = 3, n_{\mathcal{H}} = 5$ for $|\mathcal{H}| = 211$ human operators and $k_{\mathcal{M}} = 2, n_{\mathcal{M}} = 3$ for $|\mathcal{M}| = 5$ machine operators. Due to the fact that the number of machine operators is low, the probability for re-establishing the sub share $\sigma_{\mathcal{M}}$ is 30 percent for 2 bribed machine operators and 70 percent for 3 bribed machine operators. In this case bribing means that an attacker is able to gain access to the machine operator and execute some encryption and decryption functions to reproduce the sub share $\sigma_{\mathcal{M}}$. For human operators, which could be interpreted as the results for a single-folded approach, the probability for rebuilding the sub share $\sigma_{\mathcal{H}}$ for a certain user by bribing less than 5 assigned operators converges to

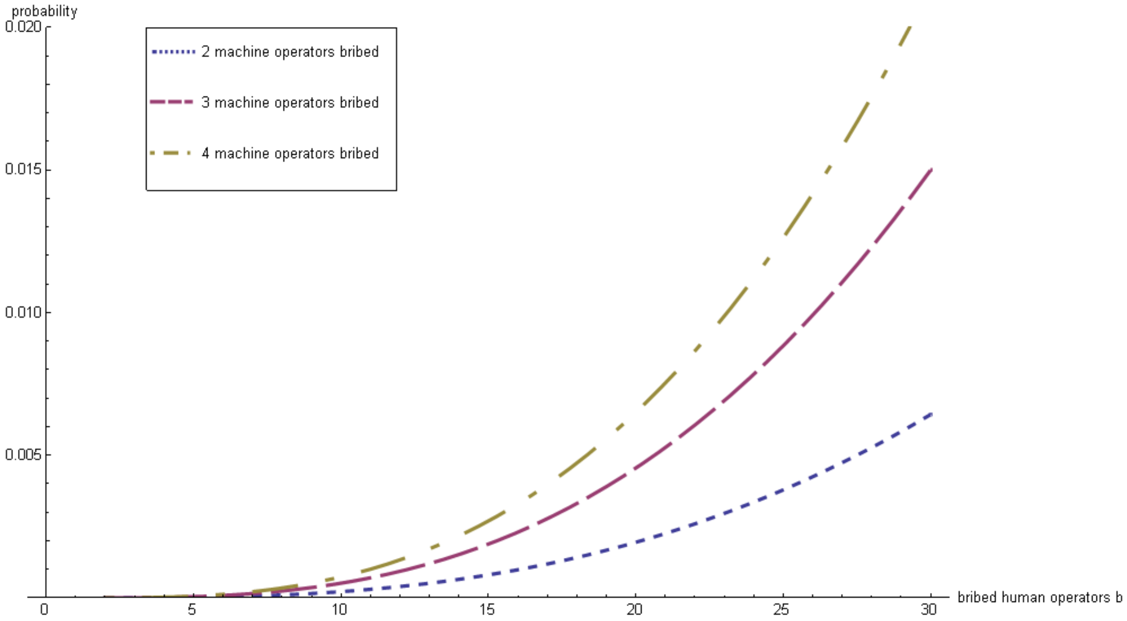


Figure 6.2: PIPE: Different combinations of bribed $\mathcal{H}$ and $\mathcal{M}$ operators

zero in this example. If an attacker bribes about 10 operators, which is nearly 5 percent of all 211 operators, the probability for reconstructing the sub share $\sigma_{\mathcal{H}}$ of a specific user is still less then 0.1 percent. Even if an attacker bribes about 30 operators, the probability would be less then 2.5 percent.

Figure 6.2 shows the result of the example for the two-folded approach, which combines human operators and machine operators as mentioned above. Even a smaller number of machine operators would decrease the probability of a successfully attack. The red line in figure 6.2, which represents 4 bribed machine operators, could be compared to the result of the single-folded approach, because the probability for reconstructing the sub share $\sigma_{\mathcal{M}}$ is 100 percent. The probability of rebuilding the inner private key of a certain user with 20 bribed human operators would be 0.7 percent with the single-folded approach and less then 0.2 percent with 2 bribed machine operators. And with the two-folded approach, the probability would be about 0.5 percent with 3 bribed machine operators.

CHAPTER 7

Cost and Performance Analysis

7.1 Costs of PIPE's fall-back mechanism

In this section, the calculation of costs for PIPE's fall-back mechanism is presented, and afterwards the costs are calculated based on the example, which has been started in section 6.2. The abbreviation used in the calculations of equations 7.1 - 7.6 are described in table 7.1.

The costs of PIPE's fall-back mechanism are split into initial costs $C_{initial}$ (cf. Equation 7.1) and current costs $C_{current}$ (cf. Equation 7.6). The estimated key recovering requests per year is based on the calculation of equation 7.3. The number of human operators and machine operators are calculated with formulas of equation 7.4 and 7.5. Furthermore, equation 7.2 presents the calculation for the time needed to set-up the system or more simply, the needed initial time $t_{initial}$.

$$C_{initial} := U * \left(\frac{n_H * C_H}{r_H^i} + \frac{n_M * C_M}{r_M^i} \right) \tag{7.1}$$

The initial costs $C_{initial}$ (cf. Equation 7.1) are calculated for each user in the system. These costs consist of the number of assigned human operators n_H multiplied by the annual costs C_H divided through the manageable requests of one human operator at set-up r^i_H and the number of assigned machine operators n_M multiplied by the annual costs C_M divided through the manageable requests of one machine operator at set-up r^i_M.

$$t_{initial} := \begin{cases} \frac{n_H * U}{r^i_H} \\ \frac{n_M * U}{r^i_M} \end{cases} \tag{7.2}$$

The initial time $t_{initial}$ (cf. Equation 7.2) is calculated separately for human and machine operators. The time needed for the set-up of the system is the number of users $|U|$ multiplied by the number of assigned human or machine operators divided through the manageable requests of one human or machine operator at set-up.

$$r := \frac{U}{sc} + U * \frac{p}{100} \tag{7.3}$$

Equation 7.3 calculates the estimated requests for key recovering. Besides the generally requests for the worn out smart cards, there are additional requests from users which have lost their smart cards.

$$|\mathcal{H}| := \begin{cases} \frac{r * k_H}{r^c_H} & \text{if } \frac{r * k_H}{r^c_H} \geq n_H \\ n_H & \text{else} \end{cases} \tag{7.4}$$

$$|\mathcal{M}| := \begin{cases} \frac{r * k_M}{r^c_M} & \text{if } \frac{r * k_M}{r^c_M} \geq n_M \\ n_M & \text{else} \end{cases} \tag{7.5}$$

Equations 7.4 and 7.5 calculate the number of human and machine operators. If the calculated number of operators is lower than the assigned operators, the minimum number of operators is the assigned operators.

$$C_{current} := |\mathcal{H}| * C_H + |\mathcal{M}| * C_M \tag{7.6}$$

The current costs $C_{current}$ (cf. Equation 7.6) are the annual recurring costs for this fall-back mechanism. These costs are calculated by the number of operators multiplied by their costs.

abbr.	description
k_H	necessary human operators
k_M	necessary machine operators
n_H	assigned human operators
n_M	assigned machine operators
sc	smart card lifetime
p	percentage of lost smart cards per year
r	estimated requests per year
r_H^i	manageable requests of one human operator at set-up
r_M^i	manageable requests of one machine operator at set-up
r_H^c	manageable requests of one human operator per year
r_M^c	manageable requests of one machine operator per year
C_H	costs of one human operator including overhead
C_M	accumulated prime and maintenance costs of one machine operator per year

Table 7.1: PIPE: Definition of attributes used in calculations

To continue the example started in section 6.2, an EHR system for 50 Mio. users is assumed, which would represent the population of England [31]. Moreover, a typical human operator with adequate education and experience would earn about 36,000 Euros [30] per year. The overhead costs for working place expenses or equipment would result a 40 percent surplus. This fact increases the total costs for one human operator C_H to approximately 50,000 Euros per year. Furthermore, an average human works 200 days a year [25], which results in about 1,600 working hours for a full-time employment. A lost smart card case can be handled by a human operator in about 30 seconds. So, she is able to handle 192,000 requests r_H^c a year to recover inner private keys $\widehat{K}_A^{-1}$. The number of requests able to be handled does not include the identification task for the patient who has lost the smart card and refers to only one of the necessary human operators $\mathcal{H}^k$.

To increase performance and decrease the cost, machine operators can be added to the system. The costs of these machine operators are divided into prime costs divided by lifetime and maintenance costs per year. The average cost for an HSM with a lifetime of 10 years and the implementation and running costs come to 3,000 Euros each year including all overhead costs. The hardware security module is able to handle 360,000 operations per hour. In best case scenario with an uptime of 99.9 percent, the HSM is able to conduct 3,150,446,400 requests r_M^c a year. This number is equivalent during the set-up process, where all assigned human operators have to encrypt the key shares $\sigma_{H_\iota}(\widehat{K_A^{-1}})$ and all machine operators have to encrypt the key shares $\sigma_{M_\iota}(\widehat{K_A^{-1}})$ of every user. At set-up time, human operators are able to handle more requests, because the encryption of the shares can be done by batch processing. Referring to this, the set-up requests are only limited by the smart cards' runtime, which is not more than one second per operation. This fact leads to a total number of 5,760,000 manageable requests r_H^i during system set-up for human operators. The typical life span is five years for a smart card, and it is assumed that the loss rate counts up to approximately seven percent a year.

To handle the system with five assigned/three necessary human operators and three assigned/two necessary machine operators under the above defined constraints, 211 human operators and 3 machine operators are needed. To increase the security, two additional machine operators should be added to the system, because with three assigned and two necessary machine operators an attacker would know that every machine operator has to hold a share of a certain user. The additional costs for two extra machine operators would only be 6,000 Euros a year. The calculated costs for this example with $|\mathcal{H}| = 211$ and $|\mathcal{M}| = 5$, the initial costs are 2,170,282 Euros, which is equal to 0.043 Euro per smart card and 10,565,000 Euros per year current costs, which is equal to 0.783 Euro per worn-out, destroyed, stolen or lost smart card.

In this example, 250,000,000 shares for the human operators and 150,000,000 machine operator requests have to be handled at the initial set-up. This amount of requests leads to a total initial time of about 41.14 working days if the shares will be randomly distributed.

7.2 Performance of PIPE's fall-back mechanism

In this section, the single- and two-folded fall-back approaches of PIPE will be investigated. Therefore, the duration of the add an actor and the key recovery process will be calculated for the normal, best and worst scenarios case based on the example given in section 6.2.

The duration of encryption and decryption functions, which are processed on a smart card, for 2048bits, is about 1 second with a 2048bit RSA key [18] and about 400 milliseconds for a 128bit AES key [36, 66]. The nCipher nShield PCI 500 HSM is able to handle the encryption and decryption of 2048bits in 1.3 milliseconds with a 256bit AES key and in 1.2 milliseconds with a 128bit AES key as shown in table 7.2. For easier calculations and to have a buffer in case of a high processor load, the average execution time of 1.5 milliseconds will be assumed for 128bit and 256bit AES keys.

The time to create an asymmetric key pair in a smart card is not easy to determine, because the smart card has to calculate a big enough prime number. This process takes no longer than 13 seconds for a 2048 bit key in more than 95% of all key generation processes [36].

As PIPE is based on a client-server architecture, the network latency and speed have to be considered as well. Therefore, an average time of 1 second for each network conversation will be assumed. This time contains all components needed for a remote procedure call (RPC) [64], for example, marshalling/demarshalling parameters and service lookups.

minutes	*128bit AES*	*Operations/sec*	*256bit AES*	*Operations/sec*	deviation
1/2	24681	823	22872	762	7%
1	49349	822	46473	775	6%
1 1/2	72999	811	69985	778	4%
2	97718	814	93094	776	5%
4	194827	812	184967	771	5%
5	245548	818	232510	775	5%
10	486779	811	463475	772	5%
∅	$\Theta(n)$	816	$\Theta(n)$	773	5%

Table 7.2: HSM: 2048bit AES encryption performance

7.2.1 Single-folded approach

In this section the duration of the add actor and report lost card workflow in PIPE for the single-folded fall-back mechanism will be investigated. As already mentioned above, the single-folded fall-back approach does not have any machine operators, but the logic (SERVICE) component has a hardware security module installed to securely handle the requests from the human operators.

Add an actor workflow

The duration of this workflow does not vary a lot between each run, because there is always the same number of operations to be done. In table 7.3 the exact number of operations needed by this workflow is presented. Next, the duration of this workflow based on the example given in section 6.2 will be calculated using the assumed time mentioned in section 7.2.

Normal case scenario: Before the example of section 6.2 is resumed, the key facts of the example will be restated. The example system is based on $k = 3, n = 5$ for $|\mathcal{O}| = 211$ operators and for about 50 million users, which would represent the population of England [31]. As the performance of the single-folded fall-back mechanism is being calculated, no machine operators are needed in this section.

The duration of the add actor workflow based on the example would be five network conversations, which takes about five seconds. Moreover, there are 10 symmetric en-/decryption processes on smart cards, which takes about four seconds, and four asymmetric en-/decryption operations on smart cards, which takes four additional seconds. Furthermore, the symmetric key and asymmetric key pair are generated on the smart card, which will take about 13 seconds. The en-/decryption with symmetric and asymmetric keys in an HSM and the key splitting with Shamir's threshold scheme could be ignored, because these operations are done in less than 100 milliseconds.

In total, the duration of the add an actor to the system workflow will be finished in less than 26 seconds with a system configuration of $k = 3, n = 5$ for $|\mathcal{O}| = 211$ operators.

Operation	*Number of Operations*
Network conversations	$n + 2$
Symmetric key generation	1
Asymmetric key generation	1
Symmetric en-/decryption (smart card)	$2 * n$
Symmetric en-/decryption (HSM)	$n + 1$
Asymmetric en-/decryption (smart card)	4
Asymmetric en-/decryption (HSM)	2
Card generation	1
Threshold scheme	1

Table 7.3: PIPE: Add an Actor - single-folded

Recover lost card workflow

The duration for the recover lost card workflow depends on the order of the answers from the operators. Therefore, the calculation will be done for the best and worst case scenario. The number of applied operations can be found in table 7.4 for the best case scenario and in table 7.5 for the worst case scenario. The abbreviation k is the number of needed operators, n is the number of assigned operators and $|\mathcal{O}|$ is the number of operators in the system.

Best case scenario: The duration of the best case scenario of the recover lost card workflow based on the example started in section 6.2 will be determined in this paragraph. There is a total number of six network conversations, which takes about six seconds. Furthermore, six symmetric and three asymmetric en-/decryption functions are applied on smart cards, which will be done in 5.5 seconds. The generation of the new smart card could be done in most cases in less than 13 seconds.

In the best case scenario, the duration of this workflow based on the example would be done in less than 25 seconds.

Operation	*Number of Operations*
Network conversations	$k + 3$
Symmetric key generation	0
Asymmetric key generation	1
Symmetric en-/decryption (smart card)	$2 * k$
Symmetric en-/decryption (HSM)	$k + 1$
Asymmetric en-/decryption (smart card)	3
Asymmetric en-/decryption (HSM)	2
Card generation	1
Threshold scheme	1

Table 7.4: PIPE: Recover lost card - single-folded - best case scenario

Worst case scenario: The longest time to finish the workflow appears when the answer of all needed operators will be the latest response of all available operators in the system. The maximum operations, that have to be executed are shown in table 7.5. In the case of the example, there are 214 network operations, 216 symmetric and three asymmetric en-/decryption operations, which have to be executed on smart cards. Furthermore, there are eight en-/decryption operations executed on a smart card and one asymmetric key generation process. In total, this workflow will be finished after 5 minutes and 15 seconds.

Operation	*Number of Operations*
Network conversations	$\|\mathcal{O}\| + 3$
Symmetric key generation	0
Asymmetric key generation	1
Symmetric en-/decryption (smart card)	$\|\mathcal{O}\| + n$
Symmetric en-/decryption (HSM)	$n + 1$
Asymmetric en-/decryption (smart card)	3
Asymmetric en-/decryption (HSM)	2
Card generation	1
Threshold scheme	1

Table 7.5: PIPE: Recover lost card - single-folded - worst case scenario

7.2.2 Two-folded approach

Contrary to the single-folded fall-back mechanism, the two-folded fall-back mechanism use not only human operators to share a secret inner private key of a user, but also machine operators as mentioned above. Before the investigations are presented, a short recapitulation of the example will be presented. The system configuration in the example is set to: $k_{\mathcal{H}} = 3, n_{\mathcal{H}} = 5$ for $|\mathcal{H}| = 211$ human operators, $k_{\mathcal{M}} = 2, n_{\mathcal{M}} = 3$ for $|\mathcal{M}| = 5$ machine operators and 50 million users.

Add an actor workflow

The difference of the single-folded fall-back mechanism is that there are additional operations needed for the machine operators. As these machine operators use a hardware security module for the cryptography functions, the total run time of this workflow would not be much higher. Though the security for the shared inner private keys will be increased a lot as shown in section 6.3.

Normal case scenario The duration of the single-folded add an actor to the system workflow is 26 seconds. In the two-folded version of this workflow, there are an additional five network conversations which take about five seconds. As with the single-folded approach, the symmetric and asymmetric en-/decryption function and the key splitting with the Shamir's threshold scheme can also be ignored, because the duration of this function would be done in less then 100 milliseconds. The total execution time of the two-folded approach would be about 31 seconds, which is not much longer compared to the single-folded approach, but provides a better security.

Recover lost card workflow

As already mentioned, the single-folded lost card recovery workflow's duration depends on the number of operators currently logged in and on the order of the answers from the

Operation	*Number of Operations*
Network conversations	$n_M + n_H + 2$
Symmetric key generation	1
Asymmetric key generation	1
Symmetric en-/decryption (smart card)	$2 * n_H$
Symmetric en-/decryption (HSM)	$3 * n_M + n_H + 1$
Asymmetric en-/decryption (smart card)	4
Asymmetric en-/decryption (HSM)	2
Card generation	1
Threshold scheme	2

Table 7.6: PIPE: Add an actor - two-folded

operators. As the machine operators are always online, in most cases, there will be $|\mathcal{M}|$ database lookups and n_M answers.

Best case scenario: In the best case scenario, there will not be much difference between the duration of the single-folded approach and two-folded approach. As the symmetric and asymmetric en-/decryption functions of the machine operators are not time consuming, there will only be one additional second for each network conversation added to the machine operators. In totally, this workflow will be executed in 30 seconds in best case.

Operation	*Number of Operations*
Network conversations	$|\mathcal{M}| + k_H + 3$
Symmetric key generation	0
Asymmetric key generation	1
Symmetric en-/decryption (smart card)	$2 * k_H$
Symmetric en-/decryption (HSM)	$2 * |\mathcal{M}| + n_M + k_H + 1$
Asymmetric en-/decryption (smart card)	3
Asymmetric en-/decryption (HSM)	2
Card generation	1
Threshold scheme	2

Table 7.7: PIPE: Recover lost card - two-folded - best case scenario

Worst case scenario: As already mentioned in the best case scenario, the duration in the worst case scenario will not be much longer than the single-folded recovery mechanism's duration. There would be an additional five seconds of network conversations in the example. The 19 en-/decryption functions executed in the HSM of the operators or the system logic (SERVICE) would not take longer than 30 milliseconds and could be ignored. The total execution time of this workflow would be 5 minutes and 20 seconds in the worst case scenario.

Operation	*Number of Operations*
Network conversations	$\lvert\mathcal{M}\rvert + \lvert\mathcal{H}\rvert + 3$
Symmetric key generation	0
Asymmetric key generation	1
Symmetric en-/decryption (smart card)	$\lvert\mathcal{H}\rvert + n_H$
Symmetric en-/decryption (HSM)	$2 * \lvert\mathcal{M}\rvert + n_M + n_H + 1$
Asymmetric en-/decryption (smart card)	3
Asymmetric en-/decryption (HSM)	2
Card generation	1
Threshold scheme	2

Table 7.8: PIPE: Recover lost card - two-folded - worst case scenario

CHAPTER 8

Conclusions and further work

Nowadays, the health care sector requires the possibility of being able to share patients' data quickly between hospital devisions and other hospitals. Medical records must be handled carefully, because the content is sensitive and could be harmful to the patient if not kept private. Therefor, governments release privacy acts to protect these records. The focus of this master thesis adheres to the privacy acts of the European Union (Data Protection Directive 95/46/EC [12]) and the United States (Health Insurance Portability and Accountability Act of 1996 [59–61]).

- Which pseudonymization approaches adhere to the current privacy laws?

To answer this research question, seven legal and seven technical requirements have been extracted from laws and needs of patients and health care providers. These requirements could be used for the future development of pseudonymization approaches. At the moment, only two out of the six evaluated pseudonymization approaches fulfill the legal requirements and most of the additional technical requirements. Therefore, only two out of the six approaches can actually be considered for use in the European Union and United States. Moreover, the results of the evaluation show that newer approaches fulfill more legal and technical requirements of the European Union and the United States.

- What are the major drawbacks of pseudonymization approaches?

During the evaluation of these six pseudonymization approaches, not only was the compliance with the applicable legal situations in the European Union and the United States investigated, but also the security of the approaches themselves. The results of the security evaluation show that there are major drawbacks in most of the systems especially in their fall-back mechanisms, which are needed to recover lost, stolen or worn out security tokens. Some approaches use a pseudonym-patient mapping list, which could very easily be abused by an insider of the system, for example a system administrator. A more secure way was presented by eGK, where all data is linked to backup security tokens. However, if both security tokens are accidentally destroyed, for example by fire, all data would be lost forever. Only two approaches suggest a solution to share the keys of the security token in the system using a threshold scheme. PIPE is the only approach which implements such a fall-back mechanism.

- How can the safe recovery of data be guaranteed, if the data has been encrypted and the patient has lost her security token?

As already mentioned, most security drawbacks were found in the fall-back mechanisms of pseudonymization approaches. In this thesis, a secure fall-back mechanism is presented which uses a threshold scheme to share a secret with operators. The advantages of a threshold scheme over a pseudonym-patient mapping list or a second card are that the operators could only recover the secret together with the appliance of a four-eye principle. Moreover, by using a threshold scheme, the secret could not be lost or destroyed, which could happen if a mapping list or backup card was used.

- Are there possible improvements of PIPE?
- Is the improved fall-back mechanism of PIPE economically justifiable?
- Is the performance of PIPE's improved fall-back mechanism fast enough?

These three research questions are answered together, because they depend on each other. As sharing a security token with system operators is much more expensive than simply using a pseudonym mapping list or a backup security token, the goal of this thesis was to improve PIPE's fall-back mechanism. As shown, it is possible to reduce the costs of the fall-back mechanism by using machine operators, for example hardware security modules.

By using such machine operators, not only have the costs been reduced dramatically, but the security of the system has also been enhanced. Furthermore, the performance of the improved system will still be suitable for the use in practice. An execution time of about 5 minutes sounds like a lot, but as this process also creates a new card, which will be sent by mail, the duration does not matter.

Further Work

Since pseudonymization approaches are often used in the health care sector, they have not only followed government laws. On the contrary, there are also additional standards to adhere to. Concrete medical software and devices have to be certified by the Food and Drug Administration to be allowed for use in the United States.

There are also standards for the exchange of medical data between different systems. Such standardized exchange processes, for example Health Level 7 (HL7), should be implemented into pseudonymization approaches to be able to exchange data between different systems.

As the fall-back mechanism of PIPE is expensive, the fall-back mechanism could be utilized as an external service, like storage at a data center. Organizations, which use PIPE or similar approaches could pay a monthly fee for the backup service. This could be much cheaper than building their own backup infrastructure. Furthermore, there are no security concerns that go along with externalizing the fall-back mechanism, because the employees of the external service provider have no possibility to recreate the shares for themselves.

Glossary

In this chapter a definition of frequently used and important key terms will be given.

- *unlinkability*: means that relations between two or more items can not be identified by pure observation of the system. Unlinkability is provided in the system with n user U if the probability that an item belongs to a specific user is $1/n$. Even an insider of the system has no possibility to gain any information related to anything [34].
- *anonymity*: means that a person is not identifiable within a set of persons A. The degree of anonymity is given by the size of the set $|A|$.
- *pseudonymization*: is a technique to hide the link between items. This link is replaced with a certain specifier which is generated by an algorithm based on encryption or hashing [23]. People who are authorized to view this link between the items can re-build it.
- *en-/decryption*: Encryption is called the process to transform plain-text into a so called chiphertext. This chiphertext cannot be easily understood by unauthorized persons. Decryption is called the process to transform the chiphertext back to plain-text.
- *hashing*: is a technique to generate unique values which is done using a cryptographic algorithm, for example MD5 or SHA-1.

Bibliography

[1] George J. Annas. Hipaa regulations - a new era of medical-record privacy? *The new england journal of medicine*, 348(15):1488–1490, 2003. URL `http://www.mcmaster.ca/ors/ethics/ncehr/2003/apr2003/1486%20NEJM%20HIPAA%20II.pdf`.

[2] Randolph C. Barrows and Paul D. Clayton. Privacy, confidentiality, and electronic medical records. *Journal of the American Medical Informatics Association*, 13:139–148, 1996. `http://www.pubmedcentral.nih.gov/picrender.fcgi?artid=116296&blobtype=pdf`.

[3] Gerd Bauer. «aktive» patiententerminals. *Datenschutz und Datensicherheit - DuD*, 30(3):138–141, 2006. URL `http://www.springerlink.com/content/w7w27581q7332257/`.

[4] David Baumer, Julia Brande Earp, and Fay Cobb Payton. Privacy of medical records: It implications of hipaa. *ACM SIGCAS Computers and Society*, 30(4):40–47, 2000. ISSN 0095-2737. doi: http://doi.acm.org/10.1145/572260.572261.

[5] Colin John Bennett. *Regulating Privacy: Data Protection and Public Policy in Europe and the United States.* Cornell University Press, 1992. ISBN: 0801480108.

[6] Bernd Blobel and Peter Pharow. Wege zur elektronischen Patientenakte. *Datenschutz und Datensicherheit - DuD*, 30(3):164–169, 2006. URL `http://www.springerlink.com/content/8113541p42777qm6/`.

[7] Detlef Borchers. Elektronische Gesundheitskarte: Die Noete der Macher. Online, 04 2008. URL `http://www.heise.de/newsticker/Elektronische-Gesundheitskarte-Die-Noete-der-Macher--/meldung/106225`.

[8] Joerg Caumanns. Der Patient bleibt Herr seiner Daten. *Informatik-Spektrum*, pages 321–331, 2006.

[9] Joerg Caumanns, Herbert Weber, Arne Fellien, Holger Kurrek, Oliver Boehm, Jan Neuhaus, Joerg Kunsmann, and Bruno Struif. Die eGK-Loesungsarchitektur. *Informatik-Spektrum*, pages 341–348, 2006.

[10] Tim Churches. A proposed architecture and method of operation for improving the protection of privacy and confidentiality in disease registers. *BMC Medical Research Methodology*, 3, 2003. doi: 10.1186/1471-2288-3-1. URL `http://www.biomedcentral.com/content/pdf/1471-2288-3-1.pdf`.

[11] Frank R. Ernst and Amy J. Grizzle. Drug-related morbidity and mortality: Updating the cost-of-illness model. Technical report, University of Arizona, 2001.

[12] 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.

[13] European Union. Data protection in the european union - citizen guide. Online: `http://ec.europa.eu/justice_home/fsj/privacy/docs/guide/guide-ukingdom_en.pdf`. URL `http://ec.europa.eu/justice_home/fsj/privacy/docs/guide/guide-ukingdom_en.pdf`.

[14] European Union, Article 29 Working Party. Working document on the processing of personal data relating to health in electronic health records (ehr), February 2007.

[15] Federal information processing standards publication. Security requirements for cryptographic modules (fips pub 140-2). Technical report, Institute of Standards and Technology (NIST), 05 2001.

[16] Fraunhofer Institut. Spezifikation der Loesungsarchitektur zur Umsetzung der Anwendungen der elektronischen Gesundheitskarte, March 2005.

[17] Lawrence O. Gostin, James G. Hodge, Jr., and Ronald O. Valdiserri. Informational privacy and the public's health: The model state public health privacy act. *American Journal of Public Health*, Vol. 91(9):1388 – 1392, September 2001.

[18] Helena Handschuh and Pascal Paillier. Smart card crypto-coprocessors for public-key cryptography. In *Smart Card. Research and Applications*, volume 1820/2000, pages 372–379, 2000. doi: 10.1007/10721064_35.

[19] Stephen Hinde. Privacy legislation: A comparison of the us and european approaches. *Computers and Security*, 22(5):378–387, 2003. doi: 10.1016/S0167-4048(03)00503-0.

[20] Gerrit Hornung, Christoph F.-J. Goetz, and Andreas J. W. Goldschmidt. Die kuenftige Telematik-Rahmenarchitektur im Gesundheitswesen. *Wirtschaftsinformatik*, 47:171–179, 2005. URL `http://www.uni-kassel.de/fb7/oeff_recht/publikationen/pubOrdner/wi_2005_171-179_hornung_telematik-rahmenarchitektur.pdf`.

[21] H.R. 84–109th Congress. Online privacy protection act of 2005. GovTrack.us (database of federal legislation), 2005. URL `http://www.govtrack.us/congress/bill.xpd?bill=h109-84&tab=summary`.

[22] Alfred Kobsa. Personalized hypermedia and international privacy. *Commun. ACM*, 45(5):64–67, 2002. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/506218.506249.

[23] Anna Lysyanskaya, Ronald L. Rivest, Amit Sahai, and Stefan Wolf. Pseudonym systems. In *Proceedings of the Sixth Annual Workshop on Selected Areas in Cryptography (SAC '99)*, 1999. URL `citeseer.ist.psu.edu/lysyanskaya99pseudonym.html`.

[24] 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.

[25] J. Monger. International comparisons of labour disputes in 2002. *Labour Market Trends*, 111:19–28, 2003.

[26] Thomas Neubauer and Mathias Kolb. Technologies for the pseudonymization of medical data: A legal evaluation. In *International Conference on Systems*. IEEE Computer Society, 2009. Received the best paper award.

[27] Thomas Neubauer and Mathias Kolb. An evaluation of technologies for the pseudonymization of medical data. *Springer Studies in Computational Intelligence*, 2009.

[28] Thomas Neubauer, Mathias Kolb, and Andreas Ekelhart. An evaluation of technologies for the pseudonymization of medical data. In *Proceedings of the ACM Symposium on Applied Computing*, 2009.

[29] Jan Neuhaus, Wolfgang Deiters, and Markus Wiedel. Mehrwertdienste im Umfeld der elektronischen Gesundheitskarte. *Informatik-Spektrum*, 29(5):332–340, 2006. doi: 10.1007/s00287-006-0102-z.

[30] Office for National Statistics (ONS). Average total income and average income tax payable: by sex, 2000/01: Regional trends 38. ONLINE, 2002. URL `http://www.statistics.gov.uk/StatBase/ssdataset.asp?vlnk=7752`. http://www.statistics.gov.uk/StatBase/ssdataset.asp?vlnk=7752.

[31] Office for National Statistics (ONS). T 04: England; estimated resident population by single year of age and sex; mid-2005 population estimates. ONLINE, 08 2006. http://www.statistics.gov.uk/statbase/Product.asp?vlnk=14508&More=Y.

[32] Organisation for Economic Cooperation and Development (OECD). Recommendation of the council concerning guidelines governing the protection of privacy and transborder flows of personal data (c(80)58/final). ONLINE, 1980. URL `http://webdomino1.oecd.org/horizontal/oecdacts.nsf/linkto/C(80)58`.

[33] Robert L. Peterson. Encryption system for allowing immediate universal access to medical records while maintaining complete patient control over privacy. *US Patent Application Publication, No.: US 2003/0074564 A1*, 2003.

[34] Andreas Pfitzmann and Marit Koehntopp. Anonymity, Unobservability and Pseudeonymity - A Proposal for Terminology. In *International workshop on Designing privacy enhancing technologies*, pages 1–9. Springer-Verlag New York, Inc., 2001. ISBN 3-540-41724-9.

[35] Klaus Pommerening and Michael Reng. Secondary Use of the EHR via Pseudonymisation. *Medical Care Compunetics 1, IOS Press*, pages 441–446, 2004.

[36] Wolfgang Rankl and Wolfgang Effing. *Handbuch der Chipkarten: Aufbau-Funktionsweise- Einsatz von Smart Cards.* Hanser Fachbuchverlag, 2008. ISBN 3446404023.

[37] Bernhard Riedl, Thomas Neubauer, and Oswald Boehm. Patent: Datenverarbeitungssystem zur Verarbeitung von Objektdaten. *Austrian-Patent, No. A 503 291 B1, 2007*, 2006.

[38] Bernhard Riedl, Veronika Grascher, and Thomas Neubauer. Applying a threshold scheme to the pseudonymization of health data. In *proceedings of the 13th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC'07)*, 2007.

[39] Bernhard Riedl, Thomas Neubauer, Gernot Goluch, Oswald Boehm, Gert Reinauer, and Alexander 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.

[40] Bernhard Riedl, Veronika Grascher, Stefan Fenz, and Thomas Neubauer. Pseudonymization for improving the privacy in e-health applications. In *Proceedings of the Forty-First Hawai'i International Conference on System Sciences*, 2008.

[41] Bernhard Riedl, Veronika Grascher, Mathias Kolb, and Thomas Neubauer. Economic and Security Aspects of the Appliance of a Threshold Scheme in e-Health. In *proceedings of the Third International Conference on Availability, Reliability and Security*, pages 39–46, 2008.

[42] Thomas C. Rindfleisch. Privacy, information technology, and health care. *Commun. ACM*, 40(8):92–100, 1997. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/257874.257896.

[43] Andreas Rottmann. CAMS dirigiert die eGK. *Datenschutz und Datensicherheit - DuD*, 30:153–154, 2006. doi: 10.1007/s02045-006-0044-3. URL `http://www.springerlink.com/content/p21417w613151055/`.

[44] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression, 1998. URL `citeseer.ist.psu.edu/samarati98protecting.html`.

[45] Thomas Schabetsberger, Elske Ammenwerth, Georg Göbel, Georg Lechleitner, Robert Penz, Raimund Vogl, and Florian Wozak. What are functional requirements of future

shared electronic health records? *Connecting Medical Informatics and Bio-Informatics*, pages 1070–1075, 2005.

[46] Bruce Schneier. Risks of data reuse. *Schneier on Security - Blog*, 2007. URL `http://www.schneier.com/blog/archives/2007/06/risks_of_data_r.html`. Last seen 19.09.2008.

[47] Bruce Schneier. Our data, ourselves. *Schneier on Security - Blog*, 2008. URL `http://www.schneier.com/blog/archives/2008/05/our_data_oursel.html`. Last seen 19.09.2008.

[48] William Seltzer. Population Statistics, the Holocaust, and the Nuremberg Trials. *Population and Development Review*, 24(3):511–552, 1998.

[49] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/359168.359176.

[50] Solveig Singleton. Privacy and human rights: Comparing the united states to europe. In *the Future of Financial Privacy*, pages 186–201, 1999.

[51] Daniel Slamanig and Christian Stingl. Privacy aspects of ehealth. In *proceedings of the Third International Conference on Availability, Reliability and Security*, pages 1226–1233, 2008.

[52] Gerhard Steinke. Data privacy approaches from us and eu perspectives. *Telematics and Informatics*, 19(2):193–200, 2002. ISSN 0736-5853. doi: http://dx.doi.org/10.1016/S0736-5853(01)00013-2.

[53] Christian Stingl and Daniel Slamanig. Berechtigungskonzept für ein e-Health-Portal. In Günter Schreier, Dieter Hayn, and Elske Ammenwerth, editors, *eHealth 2007 - Medical Informatics meets eHealth*, number 227, pages 135–140. Oesterreichische Computer Gesellschaft, 2007.

[54] Christian Stingl, Daniel Slamanig, Dominik Rauner-Reithmayer, and Harald Fischer. Realisierung eines sicheren zentralen Datenrepositories. *DACH Security*, pages 1–15, 2006.

[55] *Java Cryptography Architecture (JCA) Reference Guide.* Sun Microsystems, Inc. URL `http://java.sun.com/javase/6/docs/technotes/guides/security/crypto/CryptoSpec.html`.

[56] Latanya Sweeney. Weaving technology and policy together to maintain confidentiality. *The Journal of Law, Medicine Ethics*, 25(2-3):98, 1997.

[57] Latanya Sweeney. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

[58] Christian Thielscher, Martin Gottfried, Simon Umbreit, Frank Boegner, Jochen Haack, and Nikolai Schroeders. Patent: Data processing system for patient data. *Int. Patent, WO 03/034294 A2*, 2005. ISSN 20050043964.

[59] U.S. Congress. Health Insurance Portability and Accountability Act of 1996. *104th Congress*, 1996. URL `http://www.cms.hhs.gov/HIPAAGenInfo/Downloads/HIPAALaw.pdf`.

[60] U.S. Department of Health & Human Services Office for Civil Rights. Your Health Information Privacy Rights. ONLINE. URL `http://www.hhs.gov/ocr/hipaa/consumer_rights.pdf`.

[61] U.S. Department of Health & Human Services Office for Civil Rights. Summary of the HIPAA Privacy Rule. ONLINE, 2003. URL `http://www.hhs.gov/ocr/privacysummary.pdf`.

[62] U.S. House of Representatives. U.S. Code - Title 47 - Telegraphs, Telephones, and Radiotelegraphs - Chapter 5 - § 551. URL `http://uscode.house.gov/download/title_47.shtml`.

[63] Jan Walker, Eric Pan, Douglas Johnston, Julia Adler-Milstein, David W. Bates, and Blackford Middleton. The Value Of Health Care Information Exchange And Interoperability. *Health Affairs*, 5:10 –18, 2005. doi: 10.1377/hlthaff.w5.10. URL `http://content.healthaffairs.org/cgi/content/abstract/hlthaff.w5.10v1`.

[64] J. E. White. A high-level framework for network-based resource sharing (rfc 707). In *proceeding of the National Computer Conference*, 1976.

[65] D. Wilhelm, A. Schneider, and C. F. J. Goetz. Die neue Gesundheitskarte. *Der Onkologe*, 11(11):1157–1165, 2005. doi: 10.1007/s00761-005-0964-8.

[66] Chung-Huang Yang. Performance Evaluation of AES/DES/Camellia On the 6805 and H8/300 CPUs. *The 2001 Symposium on Cryptography and Information Security*, pages 727–730, 2001.

List of Figures

List of Tables

Listings