

Pseudonymization and its Application to Cloud-based eHealth Systems

Dissertation

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Liangyu Xu

aus

Beijing, V.R. China

Bonn, 2016

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn.

Erstgutachter: Prof. Dr. Armin B. Cremers, Bonn
Zweitgutachter: Prof. Dr. Ulrike Meyer, Aachen
Tag der Promotion: 31.03.2017
Erscheinungsjahr: 2017

Abstract

Responding to the security and privacy issues of information systems, we propose a novel pseudonym solution. This pseudonym solution has provable security to protect the identities of users by employing user-generated pseudonyms. It also provides an encryption scheme to protect the security of the users' data stored in the public network. Moreover, the pseudonym solution also provides the authentication of pseudonyms without disclosing the users' identity information. Thus the dependences on powerful trusted third parties and on the trustworthiness of system administrators may be appreciably alleviated.

Electronic healthcare systems (eHealth systems), as one kind of everyday information system, with the ability to store and share patients' health data efficiently, have to manage information of an extremely personal nature. As a consequence of known cases of abuse and attacks, the security of the health data and the privacy of patients are a great concern for many people and thus becoming obstacles to the acceptance and spread of eHealth systems. In this thesis, we survey current eHealth systems in both research and practice, analyzing potential threats to the security and privacy. Cloud-based eHealth systems, in particular, enable applications with many new features in data storing and sharing. We analyze the new issues on security and privacy when cloud technology is introduced into eHealth systems.

We demonstrate that our proposed pseudonym solution can be successfully applied to cloud-based eHealth systems. Firstly, we utilize the pseudonym scheme and encryption scheme for storing and retrieving the electronic health records (EHR) in the cloud. The identities of patients and the confidentiality of EHR contents are provably guaranteed by advanced cryptographic algorithms. Secondly, we utilize the pseudonym solution to protect the privacy of patients from the health insurance companies. Only necessary information about patients is disclosed to the health insurance companies, without interrupting the current normal business processes of health insurance. At last, based on the pseudonym solution, we propose a new procedure for the secondary use of the health data. The new procedure protects the privacy of patients properly and enables patients' full control and clear consent over their health data to be secondarily used.

A prototypical application of a cloud-based eHealth system implementing our proposed solution is presented in order to exhibit the practicability of the solution and to provide intuitive experiences. Some performance estimations of the proposed solution based on the implementation are also provided.

Überblick

Um gewisse Sicherheits- und Datenschutzdefizite heutiger Informationssysteme zu beheben, stellen wir eine neuartige Pseudonymisierungslösung vor, die benutzergenerierte Pseudonyme verwendet und die Identitäten der Pseudonyminhaber nachweisbar wirksam schützt. Sie beinhaltet neben der Pseudonymisierung auch ein Verschlüsselungsverfahren für den Schutz der Vertraulichkeit der Benutzerdaten, wenn diese öffentlich gespeichert werden. Weiterhin bietet sie ein Verfahren zur Authentisierung von Pseudonymen, das ohne die Offenbarung von Benutzeridentitäten auskommt. Dadurch können Abhängigkeiten von vertrauenswürdigen dritten Stellen (trusted third parties) oder von vertrauenswürdigen Systemadministratoren deutlich verringert werden.

Elektronische Gesundheitssysteme (eHealth-Systeme) sind darauf ausgelegt, Patientendaten effizient zu speichern und bereitzustellen. Solche Daten haben ein extrem hohes Schutzbedürfnis, und bekannte Fälle von Angriffen auf die Vertraulichkeit der Daten durch Privilegienmissbrauch und externe Attacken haben dazu geführt, dass die Sorge um den Schutz von Gesundheitsdaten und Patientenidentitäten zu einem großen Hindernis für die Verbreitung und Akzeptanz von eHealth-Systemen geworden ist. In dieser Dissertation betrachten wir gegenwärtige eHealth-Systeme in Forschung und Praxis hinsichtlich möglicher Bedrohungen für Sicherheit und Vertraulichkeit der gespeicherten Daten. Besondere Beachtung finden cloudbasierte eHealth-Systeme, die Anwendungen mit neuartigen Konzepten zur Datenspeicherung und -bereitstellung ermöglichen. Wir analysieren Sicherheits- und Vertraulichkeitsproblematiken, die sich beim Einsatz von Cloud-Technologie in eHealth-Systemen ergeben.

Wir zeigen, dass unsere Pseudonymisierungslösung erfolgreich auf cloudbasierte eHealth-Systeme angewendet werden kann. Dabei werden zunächst das Pseudonymisierungs- und das Verschlüsselungsverfahren bei der Speicherung und beim Abruf von elektronischen Gesundheitsdatensätzen (electronic health records, EHR) in der Cloud eingesetzt. Die Vertraulichkeit von Patientenidentitäten und EHR-Inhalten werden dabei durch den Einsatz moderner kryptografischer Algorithmen nachweisbar garantiert. Weiterhin setzen wir die Pseudonymisierungslösung zum Schutz der Privatsphäre der Patienten gegenüber Krankenversicherungsunternehmen ein. Letzteren werden lediglich genau diejenigen Patienteninformationen offenbart, die für den störungsfreien Ablauf ihrer Geschäftsprozesse nötig sind. Schließen schlagen wir eine neuartige Vorgehensweise für die Zweitverwertung der im eHealth-System gespeicherten Daten vor, die die Pseudonymisierungslösung verwendet. Diese Vorgehensweise bietet den Patienten angemessenen Schutz für ihre Privatsphäre und volle Kontrolle darüber, welche Daten für eine Zweitverwertung (z.B. für Forschungszwecke) freigegeben werden.

Es wird ein prototypisches, cloudbasiertes eHealth-System vorgestellt, das die Pseudonymisierungslösung implementiert, um deren Praktikabilität zu demonstrieren und intuitive Erfahrungen zu vermitteln. Weiterhin werden, basierend auf der Implementierung, einige Abschätzungen der Performanz der Pseudonymisierungslösung angegeben.

Personal information

Liangyu Xu

Education

Since 2013	Rheinische Friedrich-Wilhelms-Universität Bonn Doctoral fellow of Computer Science
09/2004 – 07/2007	Chinese Academy of Sciences, China Study of Information Security, Graduated with Master degree of Sci.
09/2000 – 07/2004	Tsinghua University, China Study of Electronic Engineer, Graduated with Bachelor degree of Sci.

Acknowledgements

The doctoral study in Germany is an unexpected period in my life. Although it was a difficult decision for me in the beginning, the three years in University of Bonn finally come to me with much pleasure and many harvests. I think this is because of those kind people helping and encouraging me, and I would like to express my sincerely appreciation to them here.

First of all, I thank Prof. Dr. Armin B. Cremers. He is so farsighted and suggested me my current research topic on the security and privacy in eHealth systems. He arranged many seminars and meetings with other researchers and provided me opportunity to broad my view and improve my research. Especially, from many times of discussions with him, I benefit not only in the knowledge, but also in the thinking manner of research and work. He also spent much of his precious time to help me revise the papers and this thesis. I must also thank for his stipend during my study and the financial support for attending several international conferences to present our work and communicate with other people.

During my research, I have got a lot of suggestions and help from Dr. Adrian Spalka, Jan Lehnhardt, Tobias Wilken. At many times of discussions with Dr. Adrian Spalka, he raised many challenges and comments from the viewpoint of practical eHealth systems. He also pointed out some shortcomings in my initial proposed schemes and reminded me to fix them. Jan Lehnhardt discussed with me about the technical details of the schemes and also raised some questions and improvement. What is more, Jan Lehnhardt revised some of my papers in English writing. Tobias Wilken has a lot of experience in software development and helped with the first implementation of the whole solution.

I would also like to express my appreciation to Prof. Dr. Meyer, who revised my dissertation carefully and gave me a lot of comments for a clearer presentation of the dissertation.

At last, heartfelt thanks to my family members. Thanks for your company and encourages during this period of hard time. Especially my wife gave birth to our son during this time. I know that she really suffered from a lot of pain and pressure although she often says she enjoys so. I love you and hope we can have a better life in the following years with our loved son.

Contents

1	Introduction	1
1.1	Motivation of the Thesis	1
1.2	Contributions of the Thesis	2
1.3	Structure of the Thesis	3
2	Background Knowledge	5
2.1	Pseudonym	5
2.1.1	Definition	5
2.1.2	Basic Requirements for Pseudonyms	5
2.1.3	Pseudonym versus Anonym	7
2.2	eHealth Systems	9
2.2.1	History of Health Records	9
2.2.1.1	EMR	9
2.2.1.2	PHR	9
2.2.1.3	EHR	10
2.2.2	Existing eHealth Systems	11
2.2.2.1	England	11
2.2.2.2	Netherlands	12
2.2.2.3	Germany	13
2.2.2.4	North America	14
2.2.2.5	Other Areas	15
2.2.3	Conclusions on the Existing eHealth Systems	15
2.3	Important Notions	17
3	Security and Privacy Issues in Different Models of eHealth Systems	19
3.1	A General Model of eHealth Systems	19
3.1.1	Overview	19
3.1.2	Basic Entities in eHealth Systems	21
3.1.3	Basic Activities in eHealth Systems	24
3.1.4	IT Infrastructure	29
3.2	Security and Privacy Threats	32
3.2.1	The Owner of Electronic Health Records (EHR)	32
3.2.2	General Attackers	33
3.2.3	Healthcare Providers	35
3.2.4	Insurance Companies	36
3.2.5	Secondary Users	37

3.2.6	Trusted Third Parties	38
3.3	A Cloud-based Model of eHealth Systems	39
3.3.1	Principles for the Cloud-based eHealth Model	39
3.3.1.1	Rule out the Fully Trusted Third Parties	39
3.3.1.2	Utilize Smart Card for Trust Computing Module	41
3.3.1.3	Adopt Cloud as Storage and Communication Media	41
3.3.1.4	Protect EHR and Messages for Security and Privacy	41
3.3.2	Refined Procedures and New Threats	42
3.3.2.1	Registration at Insurance Company	42
3.3.2.2	Visit to Doctor	42
3.3.2.3	Medicine Purchase at Pharmacy	43
3.3.2.4	Billing.....	43
3.3.2.5	Secondary Use.....	43
4	State of the Art	45
4.1	Pseudonym Schemes for General Purpose	45
4.2	Pseudonym Schemes for eHealth Systems	47
4.3	Comparison of Existing Pseudonym Solutions	48
4.4	Expected Revolutions in Pseudonym Schemes for eHealth Systems	48
5	A Novel Pseudonym Solution.....	51
5.1	Features of the Pseudonym Solution.....	51
5.2	Setup of Secret Key.....	52
5.2.1	Preparation of Setup.....	52
5.2.2	Algorithm for Setting up Secret Key	52
5.3	Algorithm for Generating Pseudonyms	53
5.4	Algorithm for Reproducing Pseudonyms	55
5.5	Authentication of Pseudonyms	57
5.6	Security Evaluation	58
5.6.1	Proof of Algorithm 2.....	59
5.6.2	Proof of Algorithm 3.....	59
5.6.3	Proof of Algorithm 6, 7, 8.....	59
6	Applications of the Pseudonym Solution in eHealth Systems	61
6.1	Application in Ordinary Healthcare Activities	61
6.1.1	Cloud-based eHealth System with Pseudonyms.....	61
6.1.2	Concealing Patients' Identities and Protecting the Security of EHR.....	63
6.1.3	Indexing EHR (Electronic Health Records) Entries	64
6.1.4	Retrieving EHR Entries.....	64
6.1.5	Authenticating the Ownership of EHR	65
6.1.6	Purchasing Medicines from Pharmacy	65

6.1.7	Potential Threats and Countermeasures	66
6.1.7.1	Cloud's Knowledge of the PIDs of Patients	66
6.1.7.2	Protecting Identities against Eavesdroppers	67
6.1.7.3	Avoiding Guessing Attacks against Secret Keys	67
6.1.7.4	Dealing with Pseudonym Collision.....	68
6.1.7.5	Trust Mode of Cloud	69
6.1.7.6	Extra Threats from Optional Algorithms.....	70
6.2	Application in Health Insurance.....	70
6.2.1	A Refined Billing Procedure	70
6.2.1.1	Setting in Billing	70
6.2.1.2	Validating Patients	72
6.2.1.3	Generating a Bill	72
6.2.1.4	Validating a Bill	73
6.2.2	Pseudonymizing Bills	74
6.2.2.1	Initialization.....	74
6.2.2.2	Generation of Pseudonym.....	74
6.2.2.3	Generation of Pseudonymous Signature.....	75
6.2.2.4	Verification of Pseudonymous Signature	75
6.2.3	Potential Threats and Countermeasures	76
6.2.3.1	Trustworthiness of Healthcare Providers.....	76
6.2.3.2	Adapting to Practical Billing Models.....	78
6.2.3.3	The loss of smart card	79
6.3	Application in Secondary Use of Health Data	80
6.3.1	Setting of Secondary Use in Cloud-based eHealth Systems.....	80
6.3.1.1	Secondary Users	80
6.3.1.2	Patient's Secrets and Certificate.....	81
6.3.2	Pseudonymized EHR from Ordinary Healthcare Activities	81
6.3.2.1	Visiting a Doctor.....	81
6.3.2.2	Getting Medicines from a Pharmacy.....	82
6.3.3	Secondary Use of Pseudonymized EHR	83
6.3.3.1	Requirements to the Pseudonyms.....	83
6.3.3.2	Searching for Participants.....	84
6.3.3.3	Contacting Target Patients for Consent and More.....	84
6.3.3.4	Responding to the Secondary User.....	85
6.3.3.5	Sending Feedback to Patients.....	86
6.3.4	Potential Threats and Countermeasures	87
6.3.4.1	Partial Encryption.....	87
6.3.4.2	Decision on Own Risk.....	87

6.3.4.3	Trustworthiness of Cloud.....	88
6.3.4.4	Other Potential Attacks in Secondary Use	88
7	Implementation.....	89
7.1	About the Implementation.....	89
7.1.1	Joint Work.....	89
7.1.2	Development Environment	89
7.1.3	Availability of the Implementation	90
7.2	Architecture of the Implementation	90
7.3	User Interfaces	90
7.3.1	Patient’s Secret Key Setup.....	90
7.3.2	Visiting Doctor	91
7.3.3	Viewing and Managing EHR.....	91
7.3.4	Purchasing Medicines	93
7.3.5	Secondary Use	93
7.4	Performance Evaluation	98
7.4.1	Evaluation on the Cloud Side	98
7.4.2	Evaluation on the Client Side	98
7.4.2.1	Performance of Simulated Smart Card by Computer Software.....	98
7.4.2.2	Performance Estimation of Real Smart Cards	100
7.4.3	Decreasing the Computation Load of Smart Card.....	101
7.5	Future Work for the Implementation	102
8	Conclusions.....	103
	Bibliography	107
	List of Figures	115
	List of Tables/Algorithms.....	116
	Appendix: Load Tests on Cloud and Database	117

1 Introduction

1.1 Motivation of the Thesis

Thanks to the fast development of information technologies, electronic health systems (eHealth systems) (Eysenbach, 2001, Oh et al., 2005) have been turning into reality and becoming a hot field in academic research. eHealth systems utilize computer and network to store and transmit the data created in various healthcare activities, e.g. making appointments, recording examinations and therapy details, purchasing medicines with prescriptions, billing health insurance companies, reusing health data for medical research. With the help of information technologies, eHealth systems greatly improve the efficiency and the quality of health care comparing to the traditional paper-based healthcare systems (Black et al., 2011). For example, in eHealth systems, a doctor is able to conveniently review a patient's previous health records created even by other doctors through a computer connected to the network instead of seeking from a large pile of paper files. The efficient sharing of the health records helps the doctor to make faster and better diagnosis and treat on the patient. Hence, eHealth systems are becoming more and more popular in the world (Borycki et al., 2012, Shu et al., 2014, Stroetmann et al., 2011, Bouamrane and Mair, 2011, Rau et al., 2010).

While eHealth systems facilitate the sharing of health data, they also bring a lot of new issues in different aspects. The first one is how to efficiently store and share the electronic health records (EHR) of numerous patients created by different healthcare providers (e.g. practices, hospitals, and mobile health sensors). Because a patient may visit many hospitals, practices and pharmacies, EHR of the patient are stored at different locations. It is challengeable to share the patient's EHR among different healthcare providers (Jin et al., 2009). Where to store EHR with huge size and how to share them with low time delay is still unresolved. The second issue comes from the security of EHR. The security refers to many aspects, e.g. the confidentiality, integrity, and authenticity of EHR (Wang and Wulf, 1997). Because EHR of patients need to be frequently accessed and updated in different healthcare activities by different people (e.g., healthcare providers, medical researchers, and even patients themselves), the security of EHR may be invaded from outside and even inside of eHealth systems. Another major issue is the privacy of patients. EHR, containing highly private information of patients, e.g. the illnesses and the identities, can be easily duplicated and disclosed over the network when they are used for different purposes (e.g. for sharing among doctors, medical research). The disclosure of these data will seriously infringe the privacy of patients (Slamanig and Stingl, 2008). How to protect the privacy of patients while preserving the easy sharing of EHR is still challengeable. The privacy of patients involves several aspects: the identity management (patients' identities should be only known by necessary entities), the health data sharing policy (patients' health data should never be disclosed to unintended people) and the consent management (patients should have full control on the uses of their health data). The security protection on the health data does not necessarily guarantee the privacy of patients. For example, although EHR are securely encrypted by advanced cryptographic algorithms e.g. AES (Daemen and Rijmen, 2013), some side-channel information in the encrypted data

(e.g., creating date/time, location, doctor's signature) may be used to deduce a patient's health status and even the identity.

According to our best knowledge, these issues are far away from being solved and remain as big obstacles to hinder the development and popularization of eHealth systems. For the time being, EHR of patients are stored at different locations (e.g., hospitals and practices) independently even with different formats (Jones et al., 2010, Drees, 2007), because different eHealth systems are used by different healthcare providers. As a consequence, it is difficult to share EHR among different healthcare providers due to many issues, e.g., the network connection and authorization problems. Some proposed eHealth clouds (Hu et al., 2012, Barua et al., 2011, Fan et al., 2011, Narayan et al., 2010, Sunyaev et al., 2010) attempt to enable patients or healthcare providers to register and upload their EHR to a public cloud for central storing and easy sharing. However, the security of EHR and the privacy of patients are fully controlled under the protection mechanism implemented by the cloud providers. However, the trustworthiness on the cloud providers is doubtful due to the business model of cloud, e.g., a public cloud may distribute the storage servers physically all over the world (Foster et al., 2008, Vaquero et al., 2008, Mell and Grance, 2011). In addition, these eHealth clouds only deal with the security and privacy issues in one special healthcare activity (e.g. only for sharing EHR among doctors), but they are not applicable or unknown to be generalized to other healthcare activities (e.g. for medical research).

Besides eHealth systems, there exist many other information systems in our daily lives. As a result, more and more private data are stored in and transmitted over them. Our privacy is suffering from higher and higher risk to be leaked or maliciously used. Unfortunately, due to the differences in the system structure and business model, it is difficult to provide a uniform solution to deal with the privacy issues in all of these information systems. However, the idea of protecting the identities by pseudonyms is generally adopted in many systems. We are curious to know what a pseudonym solution can act to protect the users' identities and further to protect the security of the private data in the information systems, especially in eHealth systems.

1.2 Contributions of the Thesis

We survey the business models of many existing eHealth systems, and analyze the current technical solutions in industry and academic research work. Based on the existing schemes, this thesis proposes a new pseudonym solution aiming to be widely used in major healthcare activities. In summary, the main contributions of the thesis are listed as follows:

- We extend the use of cloud into major healthcare activities in eHealth systems and propose a new cloud-based eHealth model. In this model, all entities in an eHealth system are connected together by the cloud. EHR can be conveniently created, shared and managed by different entities through the cloud applications. We even refine the processes of several healthcare activities to be more suitable for cloud environment. Based on this new eHealth model, various security and privacy threats from different sources are analyzed. Some corresponding countermeasures are provided, and also some open questions are raised.

- A novel pseudonym solution is proposed. The pseudonym solution can be used for storing, accessing and sharing the private data of users in the cloud. It provably protects the security of the data and enables the users to fully control their privacy. Each user generates her/his own pseudonyms and encryption keys from a secret key which is only known by herself/himself. All the private data and communications are protected by the secret key only known by the user. The secret key is the only credential to control the use of private data. Moreover, the pseudonym solution also enables users to prove their ownership of the private data without disclosing any identity information to the verifiers. Moreover, the pseudonym solution can work without a single trusted third party, which lowers the risk of disclosing the privacy of users.
- We apply the proposed pseudonym solution in several major healthcare activities to demonstrate its practicability in eHealth systems. To comply with the normal processes of these healthcare activities, some corresponding protocols for using the pseudonym solution are designed. With the new pseudonym solution, most business models of existing eHealth systems do not need to change dramatically. Instead, the new pseudonym solution can be introduced as additional features to enhance the efficiency, security and privacy for the current eHealth systems. Moreover, using the same pseudonym solution in different healthcare activities simplifies the design of security mechanism, and decreases the risk of disclosing the privacy of patients due to the weakness in a single activity.
- A prototypical eHealth system supported by the proposed pseudonym solution is implemented. The eHealth system is deployed on a PaaS (Platform as a Service) cloud. It provides several major healthcare activities involving common entities in real health care, e.g., patients, doctors, pharmacist and secondary users. These entities can create, access and update EHR by web browsers connected to the cloud. Through the prototypical eHealth system, we can experience the advantages that our pseudonym solution brings, especially in the security and privacy aspects. Moreover, the performance of our implementation and the critical algorithms in our solution are examined through simulation tests and estimations. The evaluation results indicate the practicability of our pseudonym solution under current software and hardware conditions.

1.3 Structure of the Thesis

The next chapter 2, “Background Knowledge”, firstly introduces the fundamental notions such as pseudonym and eHealth system. Especially for “pseudonym”, we explain it with some simple examples and also with the formal definition. Moreover, we briefly introduce the history of eHealth systems, and present some open problems in existing eHealth systems.

In chapter 3, “Security and Privacy Issues in Different Models of eHealth Systems”, we describe in detail various threats to the security of health data and the privacy concerns of patients. We will begin with a traditional eHealth model, and then analyze the causes and the consequences of these security and privacy threats. Afterwards, we propose a cloud-based model for modern eHealth systems. This new model addresses many shortcomings that appeared in the traditional model, but it also introduces new challenges and threats, which are further discussed.

The chapter 4, “State of the Art”, reviews literatures on existing pseudonym schemes for the purpose of general uses and for using in eHealth systems. A comparison on their supported technologies and features is made. Based on these existing schemes, we list the principles for designing a pseudonym scheme targeting at a cloud-based eHealth system.

We present a novel pseudonym solution in chapter 5, “A Novel Pseudonym Solution”. The concrete algorithms for secrets setup, pseudonym generation and the authentication protocol are formally described. Moreover, the provable security of the pseudonym solution is theoretically evaluated.

The following chapter 6, “Applications of the Pseudonym Solution in eHealth Systems”, introduces how to utilize the proposed pseudonym solution from chapter 5 in a cloud-based eHealth system. We choose several important healthcare activities in eHealth systems, and show how the new pseudonym solution can be applied to solve the security and privacy problems in them.

A prototypical implementation of a cloud-based eHealth system supported by our proposed pseudonym solution is introduced in chapter 7, “Implementation”. The implementation provides intuitive experiences on the advantages of our proposed solution. Moreover, some performance tests and estimations are also presented.

The chapter 8, “Conclusions”, wraps up the thesis by emphasizing the main insights and lists prospects of future work and open research questions.

2 Background Knowledge

2.1 Pseudonym

2.1.1 Definition

Pseudonyms, as a general notion, are widely used in many fields with different forms (Pfitzmann and Köhntopp, 2001). One major reason for this diversity is that pseudonyms are utilized under different settings where different requirements are addressed. Pseudonym schemes are typically used for hiding the real identities of individuals. For example, a writer uses a pseudonym to publish articles for various reasons. However, this kind of traditional pseudonyms is informal and artistic (Calisher, 1998).

With the appearance of computer science, pseudonyms are closely related to cryptography (Katz and Lindell, 2014), which makes pseudonyms more and more formal. Secret keys and cryptographic algorithms are used to generate and use pseudonyms. The reliability of the modern pseudonyms has been improved greatly due to the rigorousness and security of the cryptographic algorithms. Generally, a definition of pseudonym is presented as follows:

$$\text{PID} = \text{pid_gen}(\text{id}, x)$$

Here, “ x ” is a secret key; “ id ” is the real identity of a user; PID is the pseudonym of the user; and “ pid_gen ” is the algorithm for generating the pseudonym. The algorithm “ pid_gen ”, in which the cryptographic algorithms are employed, determines the security a pseudonym scheme. The secret x is usually only known by specified persons, like administrators in the system or the user her/himself. The real identity “ id ” can be any kind of identifiers (e.g. name, national identity number, birthday and address) or a combination of several identifiers. The generated PID can be used to replace the real identifiers where the privacy of the user is concerned. In many systems, each user has only one pseudonym. However, in more complex systems, a user can possess multiple pseudonyms.

2.1.2 Basic Requirements for Pseudonyms

One basic criteria to a pseudonym scheme is that the algorithm for generating a pseudonym must be one-way, i.e., pid_gen should be irreversible (or computational infeasible to compute the id of a user from PID without knowing the secret x) (Mollin, 2006).

$$\text{id} = \text{rev_pid_gen}(\text{PID}) \text{ is difficult.}$$

However, if the secret x is known, many pseudonym schemes can do the following computation to reverse from a pseudonym to the real identity.

$$\text{id} = \text{rev_pid_gen}(\text{PID}, x) \text{ is easy.}$$

Actually, the one-way property of a pseudonym scheme is not difficult to ensure, thanks to many one-way functions in cryptography, such as cryptographic hash functions (Rogaway and Shrimpton, 2004), cipher algorithm like AES (Daemen and Rijmen, 2000) and public key cryptography RSA (McCurley, 1990). By using these cryptographic functions, an irreversible `pid_gen` can be designed with high-level guarantee.

When a pseudonym scheme is used in an application with a lot of users, another basic requirement for pseudonyms is that they are collision-free (Lehnhardt and Spalka, 2011, Yoon and Kim, 2011). For example, in an anonymous bank, each user registers with a pseudonym computed from a secret key only known to the user. The pseudonym is the only identity code for all the financial transactions. Obviously, it must make sure that the pseudonyms of no pair of users collide (i.e., any two users A and B should not have the same pseudonym).

$$PID_A \neq PID_B$$

The collision-free requirement is not easy to satisfy for a decentralized system where users can choose the secret key x and identifiers separately (sometimes freely) to generate their pseudonyms, because many one-way functions in cryptography are not injective (Menezes et al., 1996). In a bad case, although two innocent users choose different identifiers and different secret x , they may generate the same pseudonym without knowing each other.

Another additional requirement to pseudonyms is “independence”, when each user has multiple pseudonyms. If a user uses different pseudonyms at different organizations, it should be impossible to deduce that these pseudonyms are from a same user (Martin-Löf, 1966).

“ $PID_1, PID_2, \dots, PID_n$ ” are independent with each other.

The independence is not difficult to guarantee, because there are many cryptographic functions that can produce pseudo-random outputs, such as hash functions (Blum and Micali, 1984). However, the independence of pseudonyms may be broken when users are able to use the pseudonyms or even credentials across different organizations. E.g., if a user presents his pseudonym generated by organization A at organization B, the organization B is able to know the user’s two pseudonyms at organization A and B. Even worse, organization A and B may collude to find out the pseudonyms of each user in order to trace the behaviors of users (Shokri et al., 2011).

The authentication is another requirement to a pseudonym scheme when users need to be validated by organizations, i.e., a user can prove the ownership of the pseudonym to a verifier without leaking the identity information (Cachin et al., 1999, Ravi Chandra and Sharad, 2006). For example, in an eHealth system, each user (patient) has many records in the server, and each record has an independent pseudonym as index. When a user wants to update a record indexed by a pseudonym at the server, the system should verify that the user is indeed the owner of the record (or the pseudonym) to avoid malicious updating. In order to protect the privacy of users, the server is not allowed to keep any particular pre-information (e.g. secret key or any registration information) of users. Moreover, the server should not get any useful information from the verification process to deduce

the identities of users. There are many traditional authentication schemes in cryptography, for example, a digital signature can be a candidate solution for the authentication process above. However, if a user uses the same certificate and secret key to generate signatures for different records, the pseudonyms in these records can be easily mapped to one user (i.e., the independence of the pseudonyms is broken), because these records are signed by the same public key (i.e. certificate) (Housley et al., 2002). Even worse, the certificate in the signature may imply some information (e.g. serial number, location, date) to deduce the identity of the user.

There are also some other requirements to pseudonyms in different application scenarios. We will introduce several specific pseudonym schemes in Section 4.1 and 4.2, and more additional features of these pseudonyms will be explained. For a pseudonym scheme which can be used in eHealth systems, there are also some special requirements. Moreover, due to the complexity of eHealth systems, the pseudonym scheme is also necessarily tuned to the procedure of existing and future business model. We will present our general viewpoints in Section 4.4.

2.1.3 Pseudonym versus Anonym

Pseudonym may sometimes be confused with anonym (Pfitzmann and Köhntopp, 2001). From the literal sense of the words, anonym means “nameless”, and pseudonym means “with fake name”. This can tell the difference between them to some degree. Formally, anonym aims at making the corresponding data totally un-linkable to the owners. It can be easily achieved by removing all the identifiers (e.g. name, gender, age, address). However, sometimes it is not necessary to do such clean removing, in order to anonymize the data. We just need to remove some chosen identifiers and ensure that each combination of the left identifiers has more than one record (Samarati and Sweeney, 1998). As an example, Table 1 shows an episode of the health records in a database. Each row in Table 1 presents one record entry from one patient. The 2nd to 6th columns are the identifiers of patients and the last column contains the concrete health records (e.g. examinations, prescriptions), which are omitted here.

Table 1: An example of health records with real identities

No.	Name	Post code	Gender	Age	Height	Data
1	John	53127	Male	68	175	...
2	Bob	53128	Male	68	175	...
3	Alice	53127	Female	60	170	...
4	Mary	53128	Female	60	170	...

To anonymize the data, we could certainly remove all columns from 2 to 6. However, when the anonymized data are going to be used in medical research, the gender and age are necessary to be kept. Actually we have a better way to anonymize the data to keep as much information as possible without destroying the anonymity, as shown in Table 2.

We remove all the names. Although we keep gender, age, height and even some post codes at row 2 and 4, the data still preserve the feature in the concept of “anonymous”.

Because each combination of the remaining identifiers has two matched rows, it makes the data impossible to be linked to the owners directly.

Table 2: An example of anonymizing the health records

No.	Name	Post code	Gender	Age	Height	Data
1	*	53127	Male	68	175	...
2	*	*	Male	68	175	...
3	*	53127	Female	60	170	...
4	*	*	Female	60	170	...

However, in many applications, we not only require the infeasibility of direct linking to the real identities, but also keep the ability to retrace the owners. For example, the health records used for medical research are preferable to enable “retrace” for consent management and feedback. The pseudonym can preserve the link between the pseudonymized data and the owners. Taking the above “anonymous” data as an example, we can use the pseudonyms of patients to replace the names as shown in Table 3.

Table 3: An example of pseudonymizing the health records

No.	Name	Post code	Gender	Age	Height	Data
1	pid ₁	53127	Male	68	175	...
2	pid ₂	*	Male	68	175	...
3	pid ₃	53127	Female	60	170	...
4	pid ₄	*	Female	60	170	...

The pseudonyms (pid_i) can not be linked to the names of patients or any other identifiers directly by medical researchers or any other curious attackers. However, patients or the administrators who know the secret keys to generate the pseudonyms can recover the linkage between real identities and pseudonyms by reversing the pseudonyms as introduced in Section 2.1.2. The application of pseudonym can bring benefits to medical research where patients’ health data are de-identified by a pseudonym scheme, because the researchers have the chance to request more data from patients and even send feedback to patients through the generators (e.g. administrators) of the pseudonyms. In Section 6.3, more details and advantages about the use of the pseudonym in medical research will be introduced.

Through the above examples, the difference between pseudonym and anonym can be clearly discriminated. In brief, they are used in different application scenarios with different purposes. Anonymity makes effort to remove chosen identifiers of the users to make the data unidentifiable, while pseudonyms use dedicated aliases to replace sensitive identifiers (like names, identity numbers) to ensure the data can not be directly linked to the real owners without knowing the secret keys for generating the pseudonyms. Being pseudonymous implies being anonymous if the replacement can prevent the direct linking from pseudonyms to real identifiers. In this thesis, we will use pseudonyms to achieve anonymity for health data with many other features.

2.2 eHealth Systems

2.2.1 History of Health Records

According to the sharing ability of the patients' health records, we denote different kinds of electrical health records as EMR, PHR and EHR. EMR are only used inside one hospital or practice; PHR are maintained by patients themselves and shared to persons specified by patients; EHR can be shared across different organizations. We will tell the difference among them in the following.

2.2.1.1 EMR

Traditionally, paper and films are the most common media for recording and managing patients' health records (e.g., diagnosis, prescriptions, familial illnesses, blood group, allergy) during their visits to healthcare providers (e.g. hospitals and practices). The traditional method is still running even in many countries and areas today. However, the difficulties in storing, retrieving and sharing the health records by paper have inspired the reformation of shifting to the electronic healthcare (eHealth) systems where electronic records are used instead (Hillestad et al., 2005, Fitzpatrick, 2000).

With the population of computers and network, healthcare providers individually began to deploy internal electronic systems in order to manage the internal resources, in which patients' health data are stored as electronic medical records (EMR) (Waegemann, 2003). Within the same healthcare provider, doctors can easily access the EMR on a computer via intranet, instead of seeking from piles of paper files. However, these EMR can only be used inside the healthcare provider that created them, i.e., EMR can not be shared across different healthcare providers due to various problems, e.g., network connection, different systems and format of records used, connection protocol and privacy of patients (Garets and Davis, 2006). For a workaround, patients have to carry their health records in the form of paper or discs across different healthcare providers.

2.2.1.2 PHR

Recently, some patient-centered health records (PHR) systems appear both in industry and academic research (Tang et al., 2006). Some emerging PHR systems supported by cloud have attracted many users, such as Google Health ¹ (retired on January 1, 2012), Microsoft HealthVault ², and ICW LifeSensor³. In the PHR systems, the health records are created (or imported from EMR systems) and managed by patients, who also are in charge of specify the access rights of different persons to their own health records. The PHR are usually stored at a publicly available server run by a third service provider in the Internet. The third service provider is also responsible for providing solutions to ensure the security and privacy protection, e.g. user management and access control.

¹ Google Health: <https://www.google.com/health/>

² Microsoft HealthVault : <http://www.healthvault.com/>

³ LifeSensor: <https://www.lifesensor.com>

PHR systems enable patients to have full ability of defining role-based access rights for individual health professionals (e.g., doctors, nurses, fitness coaches) to access their PHR. On the other hand, patients have to manage the complex access policy and need to understand their implications. Moreover, the third service providers are necessary to be trustworthy, which is doubtful to many patients. As a result, many users refuse to upload their health records to these PHR systems. At last, the PHR systems require that the users have the condition (e.g. computer and network) and knowledge to upload and manage their health records, which place obstacles to those plain users. Therefore, the PHR systems have their inherent limitations to be popularized in practice.

2.2.1.3 EHR

EMR or PHR, which can not be shared widely across different healthcare providers or can not cover large population, is gradually hindering the development of eHealth systems. Because the complexity of clinical trials continues to grow, the data sharing and interoperability become more and more important. With the comprehensive and fast knowledge of patients' anamnesis, doctors can make quick and correct decisions especially in some emergency cases; doctors from different hospitals can cooperate and share the common examinations like x-ray images; patients' health records can be easily obtained for secondary use, e.g. medical research to facilitate the development of medicines (Tang et al., 2006, Shekelle et al., 2006).

The widely sharable health records are called electronic health records (EHR) (Garets and Davis, 2006). EHR are created by healthcare providers (not patients any more like in PHR systems), and can be shared (e.g. via the cloud) with other health professionals at any other healthcare providers if authorized. The maintenance of EHR is often done through cooperating of different roles (e.g. healthcare providers and patients). There are more services such as billing and accounting with the health insurances involved. More participants (e.g., doctors, patients, pharmacists, insurance companies) will be engaged in such kind of EHR systems compared to the PHR systems. Due to the complexity in technology and benefit competition, many current EHR systems are built upon national-wide model to integrate various resources, for example, the eHealth system in Austria (Schabetsberger et al., 2006), and the Taiwan Electronic Medical Record Template (TMT) (Rau et al., 2010). Since these systems consider the practical requirements and are serving for real health care activities, more and more attention is being paid to them by the academic researchers and industrial product developers.

There are some international standards to unify the formats for storing EHR and exchanging messages among different eHealth systems. E.g., Health Level Seven⁴ (HL7) (Quinn, 1998), a non-profit organization, focuses on the development of international standards for healthcare informatics interoperability. These standards have been well accepted and adopted by many countries to build their eHealth systems to enable the exchange of EHR among different healthcare providers (Yang et al., 2006, Nagy et al., 2010). The development of information technologies also accelerates the development of EHR. E.g., the Internet technology and the cloud computing (Mell and Grance, 2011) (abbreviated as cloud) provide the fast and secure channel for sharing EHR. Especially, cloud computing pro-

⁴ Health Level Seven International (HL7).<http://www.hl7.org>.

vides the accessibility everywhere and anytime by any user through network. Thus the cloud is considered as an ideal media for storing EHR and allowing all potential users to access EHR conveniently.

While EHR systems facilitate the storing, retrieving and sharing of EHR, they also bring severe problems on security and privacy. During the frequent using and sharing of EHR across different healthcare providers, the contents in EHR are prone to be attacked and duplicated. As a consequence, the private information of patients might be illegally used or even spread maliciously. Even worse, many healthcare providers have outsourced the IT service to some third parties, which makes healthcare providers lose the full control of patients' EHR. It aggravates the risk of exposing the patients' private information. In the case of public cloud-based eHealth systems, because of the public accessibility of cloud, EHR stored centrally in the cloud suffer from ubiquitous attacks from the outside and even the inside of the cloud. An instinctive solution is to encrypt EHR in the cloud to avoid malicious access. However, the encryption may affect the easy sharing of EHR if the encryption keys are not available to the intended person in time. Moreover, if the encryption keys are not managed properly, there exists risk of disclosing the patient's privacy due to the incautious leak of encryption keys. For more discussion about the security and privacy issues in eHealth systems, please refer to chapter 3.

2.2.2 Existing eHealth Systems

2.2.2.1 England

England has a long history with eHealth. The National Programme for IT (NPfIT) in England already begun in 2002 and acted as the basis for eHealth deployments. In conjunction with this, the 2002 policy paper "Delivering 21st century IT support for the NHS: national strategic programme" was created. NPfIT has prioritized the sharing of clinical data across providers and the automatic transmission of electronic prescriptions to pharmacies. To this end, the National Health Service has allocated £12.4 billion to NPfIT to build an integrated national EHR system, which will also be used by pharmacies and laboratories. Approximately 5% of prescriptions are being transmitted electronically to pharmacies till 2008. In 2008, a pilot project was the Primary Care Summary Record program, where a summary of National Health Service patient data are held on a central database covering England (Cresswell and Sheikh, 2009). The purpose of the database is to make health data readily available anywhere that patients seek treatment, for example if they are staying away from their home town or if they are unable to provide information by themselves. Despite opposition from some quarters, by September 2010, 424 practices across at least 36 Primary Care Trusts had uploaded 2.7 million Summary Care Records (SCR). In 2010, the Health Secretary announced that the coalition government would continue with the introduction, but that the records would 'hold only the essential medical information needed in an emergency – that is medication, allergen and drug reactions'. By March 2013, more than 24 million SCRs had been created across England⁵.

⁵ <http://www.wraftonhousesurgery.co.uk/summary-care.asp>

In terms of ePrescription England has two programs for electronic prescribing in existence. One is the Electronic Prescription Service (EPS) which is directed at the primary care sector, GPs (general practices) and clinics, and synchronizes all steps from the generation to the dispatch of the prescription (Van Dijk et al., 2011). The other, ePrescribing, is aimed at institutions such as hospitals and includes a decision support component. In 2009 the Department of Health confirmed that over 500,000 prescriptions had been transmitted electronically in England. It is also known that some institutions have been using some form of electronic prescribing for over ten years. (Whitehouse et al., 2010)

On standards, England is included in the United Kingdom and its membership of the IHTSDO (International Health Terminology Standardisation Organisation). Alongside this, a Health Informatics Service Benchmarking and Accreditation Scheme was launched in 2008 to help health informatics providers and Information Management & Technology departments. Telemedicine initiatives in England are not combined under a single national programme but rather run at the local authority level. The Department of Health is currently funding three demonstrator projects, at local authority level, that aim to develop an evidence base for the use of telecare and telehealth in England. Aside from this, NHS direct, which provides health advice and reassurance on the phone as well as through an online library of medical advice, could also be considered as a form of telemedicine application (Whitehouse et al., 2010).

2.2.2.2 Netherlands

In the Netherlands most medical records are updated electronically and are no longer available in paper. A 2013 Survey from the National IT Institute for Healthcare in the Netherlands ('NICTIZ') and the Netherlands Institute for Health Services Research ('NIVEL') shows that 93% of general practitioners and 66% of medical specialists update their records primarily or exclusively electronically. There are several EHR solutions in place, for example the systems offered by ChipSoft, CSC-iSOFT and McKesson (Eijpe, 2013).

There are also several systems in place for the electronic exchange of patient data inserted in EHR. For example, at the regional local level there are systems that connect the information systems of general practitioners, GPs out-of-hours surgery and pharmacists (for example 'OZIS-ring'). There are also systems that connect data of medical specialists or other healthcare providers who are active in the same chain of care (for example for cancer or diabetes).

One of the current initiatives, launched by the Association of Healthcare providers for Health communication (Vereniging van Zorgaanbieders voor Zorgcommunicatie, (VZVZ)) is responsible for a system for the electronic exchange of medical data between healthcare providers⁶. The exchange of medical data between healthcare providers takes place via a National Switch Point (LSP) which provides a reference index for routing, identification, authentication, authorization and logging. Nictiz has been founded and mandated by the Dutch government to create and manage the national switch point, which should form the core of medical information exchange in the Dutch healthcare sector. The objective of the LSP is that any authorized healthcare practitioner is connected to the LSP, so

⁶ <https://www.vzvez.nl/page/Zorgconsument/Home>

that the latest and most relevant medical information about a patient can be obtained at any time, from anywhere in the Netherlands in a simple, secure and reliable way. The LSP, which should become the heart of the ICT infrastructure for Dutch healthcare, is supposed to regulate the exchange of health information between healthcare providers⁷.

The LSP can be compared to a traffic-control tower which regulates the exchange of patient data between healthcare providers. At this moment LSP mainly connects general practitioners, GPs out-of-hours surgery, pharmacists and a few hospitals. In January 2014 a spokesman of the VZVZ said that 75% of the general practitioners and 83% of the pharmacists are connected to the LSP. This system has the potential to be a nationwide system, but at the moment it is not (Eijpe, 2013).

2.2.2.3 Germany

We found no single approach to eHealth systems in Germany. While major healthcare software companies have attempted to create an infrastructure for physicians to exchange clinical data, these efforts have had minimal success. Due to security concerns, many physicians prefer to store patient records on computers that are not connected to the Internet. Despite this obstacle to eHealth systems adoption, there are currently two pilot projects focused on providing health information exchange capabilities for providers. The first one, called “D2D” (www.d2d.de) is a secure communication standard to exchange billing information and patient data. About 2300 (less than 2%) of German physicians in private practice are participating in the D2D pilot program, which conducts approximately 70,000 data transactions per month. The second program, “Vita-X” offers EHR and supports provider-to-provider exchange. While Vita-X is also in its infancy, its use is expected to become more widespread, as it is provided by the same vendor that supplies EHR systems to 50% of the German private practices (Jha et al., 2008).

The most promising approach to eHealth systems in Germany is the electronic health insurance card. As of October 2011, patients in several German states have been testing a new national electronic card. As well as being used to reimburse health costs, it offers users access to their health records online, at a national level. The aim is for the Elektronische Gesundheitskarte, or eGK (in German) to stand at the heart of a network of e-health services, delivered locally. The project has been underway since 2004, and has involved a number of setbacks. It was even brought to a halt in 2010, due to the concerns of many doctors and IT security specialists, regarding access to patients’ health data. The first version of the eGK, launched in 2006, stored medical details directly in the chip of each individual’s card. Deemed too risky, the project was abandoned, then relaunched in 2010 with a new format – by default the card now contains only social security and insurance administrative data, enabling medical costs to be reimbursed⁸.

The eGK will replace current, non-secure electronic health insurance cards. It has also been designed so that it can be used for several other optional e-health services, some of which are still in development, and all of which will be subject to the patient’s consent⁹:

⁷<https://dutchhealthcare.wordpress.com/2011/04/05/the-rise-and-fall-of-the-national-ehr-initiative-in-the-netherlands/>

⁸ <http://esante.gouv.fr/en/the-mag-issue-7/germany-and-the-challenges-rolling-out-e-health-a-large-scale>

⁹ https://www.gematik.de/cms/media/infomaterialpresse/_Broschuere_Englisch_2.pdf

- E-prescriptions: patient cards will be able to transmit prescriptions electronically to pharmacies with the necessary equipment;
- Emergency medical data: allergies, intolerances, current treatment, organ donation information, and the details of the patient's doctor can all be registered directly on the card, so that they can be easily accessed in the case of an accident;
- Treatment history and any treatment currently underway;
- A messaging and document sharing service for doctors, for discharge letters, notes, X-rays with reports, and test results, following identification via the "Elektronischer Heilsberufsausweis", the German equivalent to France's healthcare professional card (CPS);
- Access to the patient's electronic medical record if it exists. Each regional government is responsible for rolling out these records.

2.2.2.4 North America

Both Canada and the United States have experienced increases in their adoption rates of eHealth. More specifically, 2012 adoption statistics reveal that the electronic medical record adoption rate in the United States is 69% and in Canada it is 57% (Borycki et al., 2012).

In the United States, it seems that there is no nationwide eHealth system. However, the 2009 Health Information Technology for Economic and Clinical Health (HITECH) Act authorized incentive payments through Medicare and Medicaid to health care providers that use certified electronic health record (EHR) systems to achieve specified improvements in care delivery. Eligible Medicare and Medicaid physicians may receive incentive payments over 5 years, starting in 2011, if they demonstrate that they are using a certified EHR system that meets 15 Stage 1 Core Set objectives and 5 of 10 Menu Set objectives. A federally funded regional extension center (REC) program was created to provide physicians with assistance in purchasing and implementing EHR systems, training staff, and addressing how they use EHR systems when they see patients. The REC program seeks to support 100,000 primary care providers, with particular emphasis given to practices with fewer than 10 clinicians and to clinicians who work in settings that tend to serve uninsured, underinsured, and medically underserved populations. As a result, in 2012, 71.8% of office-based physicians reported using any type of EHR system, up from 34.8% in 2007. In 2012, 39.6% of physicians had an EHR system with features meeting the criteria of a basic system, up from 11.8% in 2007; 23.5% of office-based physicians had an EHR system with features meeting the criteria of a fully functional system in 2012, up from 3.8% in 2007 (Hsiao et al., 2014).

In the U.S., much of the focus on sharing of data between providers has been organized around Regional Health Information Organizations (RHIOs). These entities have already been started in many regions. RHIOs are generally nonprofit regional organizations whose primary aims are to convene healthcare providers with the hope of initiating health data exchange. Preliminary reports suggest that only a small number (fewer than a dozen) have begun to exchange clinical data. Furthermore, the financial sustainability of RHIOs, a critical factor in their long-term success, remains unknown (Protti, 2007).

In Canada, the most promising focus on eHealth systems is Infoway. Established in 2001, Infoway is an independent, nonprofit organization funded by the federal government. It also supports various projects on eHealth and sets up a lot of standards for the interoperated EHR. Across Canada, EHR are at various stages of implementation and maturity according to provincial/territorial strategies and priorities. As of February 2014, there were 11 provinces and territories reporting active EHR users. Clinicians across Canada are increasingly using EHR systems to support the care of their patients. In 2006, there were approximately 7,600 users of the electronic health record. By 2014, that number had increased significantly to over 62,000 users across Canada – an increase of more than 700 percent¹⁰.

2.2.2.5 Other Areas

Australia and New Zealand have computerized patient administration systems and many use laboratory results reporting. However, computerized documentation is limited to electronic discharge summaries which are sent directly from hospitals to general practitioners and there is little to no electronic prescribing in the hospital setting (Jha et al., 2008).

Current eHealth systems in Both Australia and New Zealand allows GPs with EHR to automatically download pathology reports and imaging reports from a variety of public and private diagnostic sectors. In addition, both countries also have national immunization registries (and Australia has a cervical smear registry, as well), which can be fed and accessed electronically. Finally, in both nations, hospitals are increasingly sending discharge summaries electronically to GPs, who in turn are sending referrals and other communications to hospitals and specialists electronically. Some experts suggest that the lack of a single national identifier code has hindered eHealth systems in Australia to a substantial degree. New Zealand, by contrast, which does have a single consumer health identifier, may have an easier time creating a national HIE program (Jha et al., 2008, Grant, 2012).

There has been some implementation of eHealth systems in both Australia and New Zealand with more likely to come. In Australia, the National E-Health Transition Authority (www.nehta.gov.au) has undertaken substantial planning for HIE implementation. In addition, a New South Wales pilot project is encouraging hospitals and community providers to exchange clinical data for some 50,000 patients (Grant, 2012).

2.2.3 Conclusions on the Existing eHealth Systems

In previous Sections, some existing eHealth systems in the world are generally introduced, and there are also a lot of undergoing eHealth events in academic research. We are not going to get into the details of these eHealth systems, because each of them is a very complex system not only in technologies, but also in policies and laws. We draw some conclusions on these existing eHealth systems based on our survey and understanding.

¹⁰Annual Report 2013-2014 - Infoway Connects: https://infowayconnects.infoway-inforoute.ca/index.php/component/docman/doc_download/2175-annual-report-2013-2014

- eHealth Systems Are More and More Popular

Because eHealth systems have many advantages for health care compared to the paper based health systems, it is a common consensus that eHealth systems will dominate in the healthcare services in the near future. More and more companies, governments are paying attention, investing resources to the development of eHealth systems.

- eHealth Systems Will Cover More Activities

In current eHealth systems, healthcare activities are still limitedly supported. For example, some eHealth systems only provide EHR storing service, without electronic prescription support. The absence of full support to healthcare activities will greatly decrease the quality of daily health care, even conceal the advantages brought by eHealth systems. Nevertheless, with the development of technology and increase of patients' demands, we believe that the future eHealth systems will cover more activities to serve the health care in our daily lives perfectly.

- Long Way to Go with the Interoperation

It is a fact that different countries have been developing their own eHealth systems, and even in a same country multiple eHealth systems are being used with different support technologies. As a result, the interoperation among different eHealth systems is an urgent question, because patients are becoming more and more mobile, e.g., moving from one doctor to another, from one hospital to another, from one city to another, and even from one country to another. Interoperation is also necessary for secondary use of EHR based on the statistics on a great amount of health data. EHR collected from different eHealth systems must be reformed to a common format to be analyzed. It is urgent to set up and adopt some international standards for the storage and exchange of health data.

- Much to Do with Security and Privacy

Because the health data are electronically stored and transmitted across healthcare providers, the security of the health data and the privacy of patients will suffer from serious threats. Some technologies and policies regarding security and privacy have been implemented in current eHealth systems. However, when eHealth systems cover more activities, these current technologies and policies may become weak or useless. The security and privacy issues are being concerned by more and more people. They will be the critical factors to the development of the eHealth systems. Thus they should be considered carefully beforehand and it is valuable to adopt some long-term effective technologies and policies to protect the security and privacy.

2.3 Important Notions

EHR (Electronic Health Records). Each patient owns one set EHR, which include all the health records during her/his past visits to the doctors

EHR Entry. A patient's EHR include many entries, each of which is created during one visit to a doctor. This entry can include any information of this visit, for example, date, examination, diagnose, prescription and so on.

Security of EHR. A patient's electronic health records (EHR) stored in server and transfer in network should guarantee the following security requirements: Confidentiality, Integrity, Authenticity, Availability.

Confidentiality. The information in EHR is not available or disclosed to unauthorized roles. This is often realized by encryption (e.g. AES) (Daemen and Rijmen, 2013). The encryption keys are protected from the unintended roles.

Integrity. Any manipulation to the EHR should be detected by the EHR users (e.g. doctors, patients). Integrity is often realized by HMAC (Krawczyk et al., 1997) and digital signature (e.g. RSA) (Shamir, 1985).

Authenticity. The owners of EHR (i.e. the patients) have the ability to prove the ownership of the data, while others are not able to impersonate the data owners (Needham and Schroeder, 1978).

Availability. The EHR should be easily available to intended roles. For example, a doctor should be able to view a visiting patient's EHR easily and comprehensive.

Privacy of patients. privacy of patients involves several aspects: the identity protection (i.e. patients' identities should be carefully used and never be disclosed to unintended roles); the health data sharing policy (i.e. patients' health data should never be disclosed to unintended people including the real identities); and the consent management (i.e. patients should have full control on the uses of their health data).

Hash function. If not specified, it means cryptographic hash function (e.g. SHA-2) (NIST, 2002), which must resist three kinds of attacks: collision, preimage and secondary preimage (Rogaway and Shrimpton, 2004).

Pseudonym, pseudonymous and pseudonymity. A pseudonym is a special kind of identifier which can not be linked to the user's real identity. Being pseudonymous is the state of using a pseudonym as ID. Pseudonymity is the use of pseudonyms as IDs (Pfitzmann and Köhntopp, 2001).

Anonymous and anonymity. Anonymity is the state of being not identifiable within a set of users. A user is anonymous when he/she does not differ from any other users in the system (Pfitzmann and Köhntopp, 2001).

Smart card. A smart card is composed a plastic card and the inner circuit. On the surface of the card, the photo of the holder can be printed. The inner circuit often can function for

encryption/decryption/signature and as protected storage for secret keys. If not specified, we mean the patient's smart card.

Signature. If not specified, it means digital signature with digital certificate and secret key (Housley et al., 2002).

Fully /Semi- trusted third party. A fully trusted party masters all the necessary information to disclose the patients' private data (e.g. EHR and identities), and thus it must be fully counted on to behave as expected (e.g. will not corrupt or be compromised). A semi-trusted third party only possesses limited information of patients, and thus its dishonest behavior or being compromised will not disclose the private data of patients, although that may disturb the normal process of healthcare activities.

3 Security and Privacy Issues in Different Models of eHealth Systems

3.1 A General Model of eHealth Systems

3.1.1 Overview

Due to the different structures of existing eHealth systems, it is difficult to analyze and compare the security and privacy levels in a fair way. We notice that almost all eHealth systems use some common elements (e.g. basic roles, healthcare activities), and they suffer from the similar threats on security and privacy. These common elements are also essential to every eHealth system. In this Section we will firstly introduce a common model of eHealth systems. Then basing on this common model, we will analyze various threats to security and privacy in a basic eHealth system in Section 3.2. It is valuable and necessary to set up a general model for eHealth systems due to the following reasons.

- Denote Common and Essential Elements from Different eHealth Systems

We have pointed out above that existing eHealth systems have different notations and different properties for the same element. However, we find that some common elements are very essential to a modern eHealth system. If we can list and denote them as uniform annotations, it will be helpful to describe a new eHealth system and the design of any technical solution (e.g. privacy and security solution) which is applied in the new eHealth system.

- Create a Reference for the Undergoing and Future eHealth Systems

There are still a lot of countries and areas lacking an eHealth system, and they are behaving differently in health care compared to those countries with developed eHealth systems. It is urgent for these countries to set up such an eHealth system to improve the quality and efficiency of health care. This general model can be a valuable reference to them. Moreover, some existing eHealth systems are undergoing revolutions in terms of business model and technical solutions (e.g. adding new functions to the current systems). This basic model can also be referenced by them.

- Act as a Common Platform where Security and Privacy Are Evaluated

Security and privacy issues are great challenges for each eHealth system. There are many solutions responding to the security and privacy in industry and academic research. Because these solutions may focus on different fields of eHealth systems, it is difficult to compare, evaluate and integrate them together. In a practical eHealth sys-

tem, because security and privacy are global problems, any successful attack or infringement to the health data in a single field may cause the failure of the whole system. So a qualified solution for security and privacy should be a solution set which can secure all essential fields and can even potentially support some upcoming functions to eHealth systems. With the help of this general model, we can evaluate a solution on security and privacy in a common setting, and locate which fields are addressed. The compatibility and interoperability of the solution with existing technologies can also be examined accordingly.

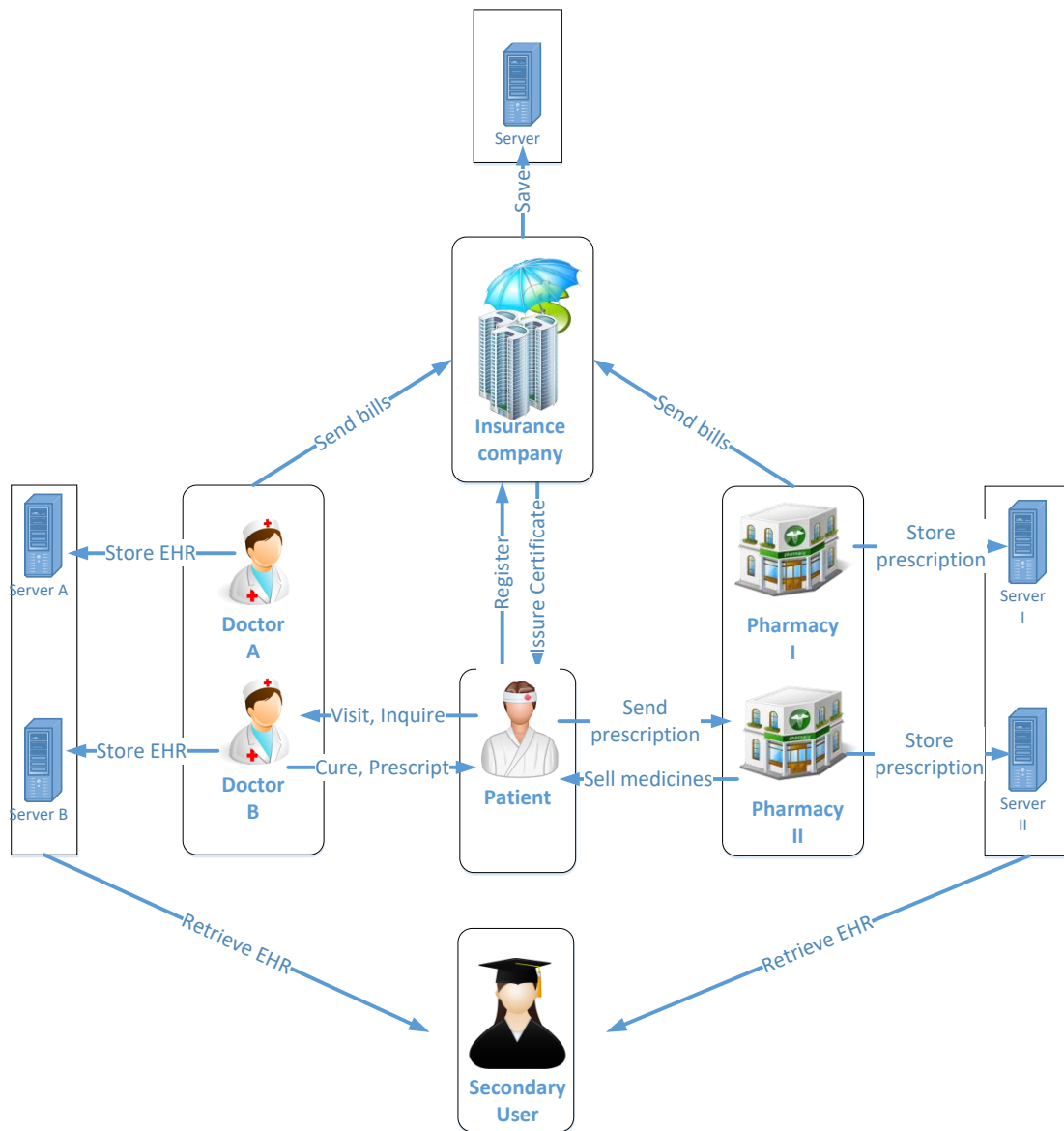


Figure 1: A general model of eHealth systems

As shown in Figure 1, we present a general model of eHealth systems. This model consists of the basic entities (denoted by round-corner rectangles), IT infrastructure (denoted

by rectangles), and healthcare activities (denoted by arrows). We will describe these elements in the following Sections one by one. Note that these elements may be named into different annotations in different eHealth systems, and there are more or less elements in different eHealth systems.

3.1.2 Basic Entities in eHealth Systems

- Patient

Patient is one of the most important roles in eHealth systems. A patient can be anybody in society and even anonymous in some traditional healthcare systems. However, in an eHealth system, patients' identities need to be uniquely represented for different reasons. When a doctor wants to access a patient's previous EHR, these previous EHR must be connected together with a unique identification, otherwise, other patients' EHR may be wrongly thought as belonging to this patient. Moreover, a health insurance company also needs to distinguish different customers by the identifications.

Considering the privacy of patients, it is not easy to get a simple solution to represent patients' identities. Some currently used methods in existing eHealth systems have potential privacy issues. For example, using patients' identity numbers or any unique identifiers will easily disclose patients' identities (e.g. an identity number may easily be mapped to name, address etc.); using an insurance card number may also leak the privacy to unintended persons (refer to Section 6.2); using digital certificates from a trusted party may also potentially be attacked (refer to Section 3.2.6).

The privacy issue is mainly concerned by patients in eHealth systems. The disclosure of a patient's health information may cause severe consequences, e.g. cause negative impact to the employment of the patient. Note that, not only some intentional attackers may infringe patients' private information for their various kinds of purposes, but also some normal use of patients' health data may leak the privacy of patients unintentionally.

- Doctor

Another important role in eHealth systems is the doctor. A doctor can be any professional physician in a small practice or in a big hospital, and also can be any medical care personnel such as a nurse, or laboratory technician. Doctors usually need to be certified by some trustworthy organizations or government departments. They have to get licenses (e.g. digital certificates) in paper or in digital form. So they are typically identified by their real identities. Because they can directly contact to the EHR of patients, they play an important role to protect the security of the health data and the privacy of patients. Generally, doctors have little to no privacy problem as they are willing to be known by the public e.g. for advertisement reason. However, some doctors may be unwilling to disclose the amount of patients they have treated. If patients' EHR are public available to everybody (e.g. in the eHealth model introduced in Section 6), this privacy issue of doctors should also be considered. This can be achieved by trusted third parties (i.e. the certificate authority issue to doctors the certificate in which the real identities of doctors are not included). For the simplicity, we will not consider the privacy of doctor further in this thesis.

In most current eHealth systems, doctors are considered to be fully trustworthy. It is true to some degree. We also utilize the trustworthiness of doctors in our proposed security and privacy solution. Meanwhile, we think that doctors may behave dishonestly in some occasions. E.g., they might disclose patients' health data intentionally or unintentionally, and they might cheat for illegal benefit from insurance companies by sending fake bills. We will propose solutions to detect and avoid these malicious behaviors of dishonest doctors (refer to Section 6.2.1).

- Pharmacy

A pharmacy can be a substantial or online shop where pharmacists sell medicines to patients according to prescriptions. In some countries, medicines are sold by hospitals or doctors to patients directly. However, in most developed countries and regions, pharmacies exist independently to healthcare providers. To be involved in eHealth systems, pharmacies also need to be certified by some trustworthy organizations or government departments, and get certificates in paper or digital form. Similar to doctors, they are often identified by their real identities and have no own privacy problem. However, they are able to read the prescriptions of patients and potentially leak the private data of patients. Moreover, like the dishonest doctors, a pharmacy might also cheat insurance companies to get illegal benefit by sending fake bills. Thus, not all pharmacies can be considered to be fully trustworthy. There should be countermeasures to prevent pharmacies from disclosing the private information of patients and other malicious behaviors.

- Health Insurance Company

Insurance companies are more and more involved into eHealth systems in many countries. Especially in many European countries, a valid health insurance which can cover the expenses of health care is mandatory for every citizen. Although in many countries and eHealth systems, insurance companies are still absent or run offline, we list insurance company as an essential role connected with other roles in eHealth systems. Insurance companies must also be certified by some trustworthy organizations or government departments. They have to get certificates in paper or in digital form. So they are often identified by their real identities and have no privacy problems of their own. However, insurance companies have full knowledge about each customers' health data, and thus they impose high risk to the security of the health data and the privacy of patients.

In most eHealth systems, insurance companies are considered as trusted, and they always promise to keep the privacy of patients when they sign contracts with their customers. However, we argue that it is not necessary for insurance companies to obtain full EHR of the patients. We will propose a solution to prevent insurance companies from knowing too much (just necessary information) about patients and decrease the risk of disclosing the privacy of patients (refer to Section 6.2).

- Secondary User

Besides the ordinary healthcare activities (e.g. visiting doctor, purchasing medicines), health information is also used for secondary purposes such as health system plan-

ning, management, quality control, public health monitoring, program evaluation, and medical research. Secondary users who reuse patients' health data also need to be certified by some formal organizations or government departments. They have to get certificates in paper or in digit form. So they are often identified by their real identities and have no own privacy problem.

Although the health data are usually "de-identified" or "anonymized" before they are used for secondary purposes, secondary users are still able to deduce the identities of the participating patients by the remaining insensitive identifiers (e.g. for various reasons, some identifiers such as age, gender are kept). Especially, when a secondary user wants to contact the participants to get more information and consent, or even send feedback to them, the secondary user may have a chance to know the real identities of the participants (e.g. email address, telephone number). Thus, secondary users also bring potential risk of disclosing the private data of patients.

In this thesis, we apply the pseudonym solution to protect the privacy of patients against secondary users while supporting full communication during secondary uses, e.g. de-identification, consent management and sending feedback (please refer to Section 6.3 for details).

- Trusted Third Party (TTP)

In current eHealth systems, trusted third parties are commonly used. In Figure 1 we did not draw them, because they play different roles in different eHealth systems and thus exists in different processes. We divide the trusted third parties into two categories according to the trust level on them.

The first kind of trusted third parties is powerful and acts as the trust base of the system. This kind of trusted third parties must be required as fully trustworthy because they have too much power. They play an important role to protect the security of the system and privacy of patients. This is actually the most common case of trust third parties in current eHealth systems. For example, a trusted third party manages the identities of patients and provides the secret keys for encryption and authentication on EHR. Unquestionably, the fully trusted third parties can simplify the system design and deployment, and they can guarantee the security of the system if they never do evil and are strong enough against all kinds of attacks. However, the existence of such a powerful trusted third party brings potential risk of disclosing of the privacy of patients. They are able to infringe the privacy of patients easily and seriously (e.g. to read the private information of every patient, to impersonate a patient) if they want to, because malicious insiders (e.g. the administrator of the database) may appear in the trusted third party. Moreover, the trusted third party may be compromised by some outside attackers and the private information stored may be stolen.

The second category of trusted third parties is less powerful and only needs to be "semi-trusted". "Semi-trusted" means that the third party is not able to disclose the privacy of the users, thus the security of system does not rely on the third party. However, it should be somewhat trusted to behave honestly, otherwise, the normal process of the system may run into trouble. A semi-trusted trusted third party plays an important role in the security of eHealth systems, but it just knows some partial information of

patients and is unable to obtain any further private information of the patients. Thus, the patients do not have to trust that, a semi-trusted TTP protects their private data. Instead, a semi-trusted TTP may behave dishonestly and may be compromised. The dishonest behavior or compromising will not lead to the disclosure of patients' private data, although their dishonest behaviors may interrupt the normal process of the whole system. For example, if a trusted third party in charge of authentication refuses to work or responds wrongly, the healthcare activities may be postponed or disordered. In some rare cases, the semi-trusted third parties might collude with other entities in the eHealth system to disclose the privacy of patients. For example, a trusted third party that issues certificates to patients may collude with doctors or insurance companies to decrypt all the private health data and the real identity of a patient. Nevertheless, compared to the first category of fully trusted third parties, they are much less powerful and thus the risks of disclosing the private data of patients decreases.

In our proposed solution, we assume trusted third parties as semi-trusted. This is also a trend in modern eHealth systems due to the strong demand on security and privacy.

3.1.3 Basic Activities in eHealth Systems

- Patient Registers at an Insurance Company

As we have introduced in Section 3.1.2, each patient needs to contact an insurance company to sign a health insurance contract as shown in Figure 2. Although in Figure 1 and Figure 2, only one insurance company is depicted for the sake of simplicity, there can be many insurance companies in a practical eHealth system, and one patient can possess multiple health insurances.

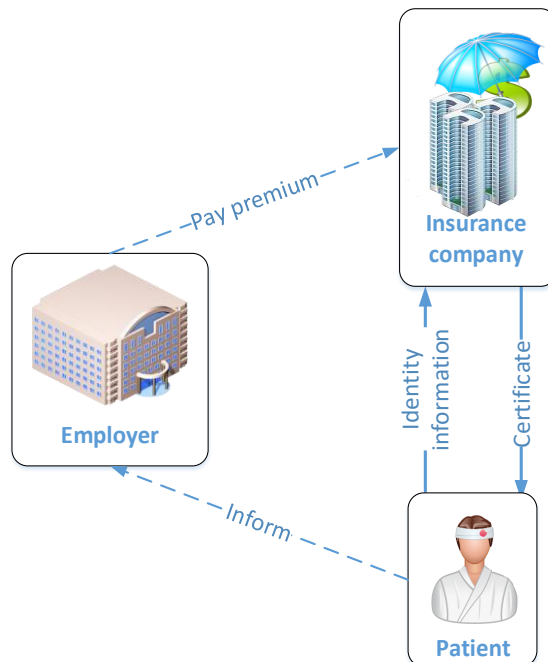


Figure 2: Patient registers at an insurance company

In the registration process, the real identity of the patient is inevitably known by the insurance company, because the insurance company usually needs to evaluate the patient's income, health status and so on. Especially, in many countries, the premium for the health insurance is indirectly paid by the employers of patients. Each patient has to inform the insurance company about his employment. As a result, the patient's identity is difficult to keep as a secret against the insurance company.

The insurance company issues a certificate (e.g., a digital certificate in a patient smart card) with signature (e.g. digitally signed by the private key of the insurance company) to the patient. Besides that, the photo of the patient is often printed on the certificate (a card or paper) to avoid the abuse of the insurance. Usually, the patient is assigned a registration reference number, which the insurance company can use to look for the patient's real identity and index the patient's data. The reference number is enclosed in the certificate issued to the patient. Other persons except the insurance company are unable to map the reference number to the real identity of the patient. Thus, the patient's certificate is anonymous to other persons. The registration procedure is important for the billing detailed in the following paragraph "Billing".

- Patient Visits a Doctor

A patient can visit a doctor (or other healthcare providers) after he/she has registered at an insurance company. The patient has to present the certificate to the doctor in order to prove the insurance coverage. The doctor can validate the certificate by checking the photo of the patient and signature of the insurance company. After that, the doctor diagnoses the patient, treats the patient, and at last writes down a prescription for the patient if necessary. In practice, the procedure of visiting doctor is much more complex as shown in Figure 3. The patient may be transferred to different medical personnel in the hospital, and there are more data created by them, besides the doctor's record on diagnosis and prescriptions. E.g., the laboratory experimenter may create examination data which are uploaded to the internal server of the hospital to be viewed by the doctor and other coworkers. For the simplicity, these medical personnel are represented by "Doctor", and all data of one patient at the internal server will be concentrated together as the health records of the patient.

In current eHealth systems, the identities of patients are often known by doctors, because doctors need to set up long term EHR (anamnesis) for patients and keep contact with patients. EHR are usually organized (or indexed) according to the identities of patients. Therefore, if a patient visits a doctor for the second time, the doctor can quickly find out the previous health information of the patient for faster diagnosis and better treatment. Since doctors master all the private health information of patients (including the identities), doctors are assumed as fully trustworthy to preserve the privacy of patients.

In practice, there are laws to regulate doctors' behaviors on patients' health data only for internal uses, and doctors have their moral faith to protect the privacy of patients. However, we argue that the identities of patients are not necessarily known by doctors. The health data of patients can be anonymous to doctors. As a result, it decreases the risk of disclosing the privacy of patients by doctors (sometimes the disclosure is unintentional).

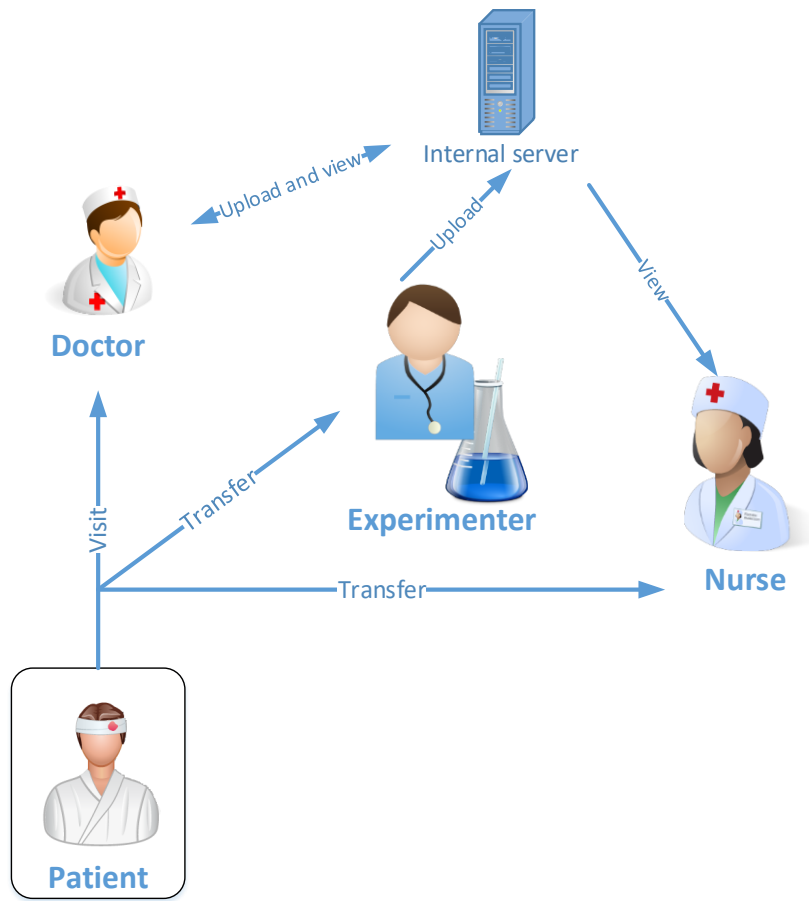


Figure 3: A patient visits a doctor in hospital

- Patient Gets Medicine from a Pharmacy with Prescription

After a patient gets a prescription (refer to Section 6.1.6 for more details about the format of a prescription) from a doctor, he/she can purchase the medicines listed in the prescription at a pharmacy (certainly patients can buy medicines without a prescription, which can be dealt with other supplementary solutions). The patient has to present the certificate from the insurance company to the pharmacist in the same way as when visiting a doctor. Then the prescription (e.g. the electrical prescription in patient's smart card) must be validated by the pharmacist to check that it is truly prescribed by a doctor. After these successful validations, the medicines are sold to patients.

For the same reason as discussed in the last paragraph, it is unnecessarily for the pharmacies to know the identities of patients during the medicine purchasing. However, in current eHealth systems, the prescriptions often include the identity information of patients for the ease of validating. As a result, a pharmacy can easily record the patients' identities and prescriptions. From the prescriptions, the pharmacy can somehow infer the patients' health status and illnesses. This potentially imposes risks with respect to disclosing the private information of patients by the pharmacists.

- Billing

After a patient left a doctor or a pharmacy, a bill including the expenses and the details of the health care administered to the patient will be sent to the insurance company by the doctor or pharmacist to get paid as shown in Figure 4(a). In the bill, the patient's registration reference number in the certificate from the insurance company will be enclosed. After the insurance company receives the bill, corresponding fees will be paid to the bill senders if the following validations succeed. The insurance company firstly examines the origin of the bill, i.e., the bill indeed originates from a registered customer to avoid abuse of insurance by checking the reference number and the signature of the patient enclosed in the bill (if available, as introduced in Section 6.2). Moreover, the signature of the bill sender must be checked in order to ensure the truth of the health care, i.e., the bill was sent by a valid bill sender.

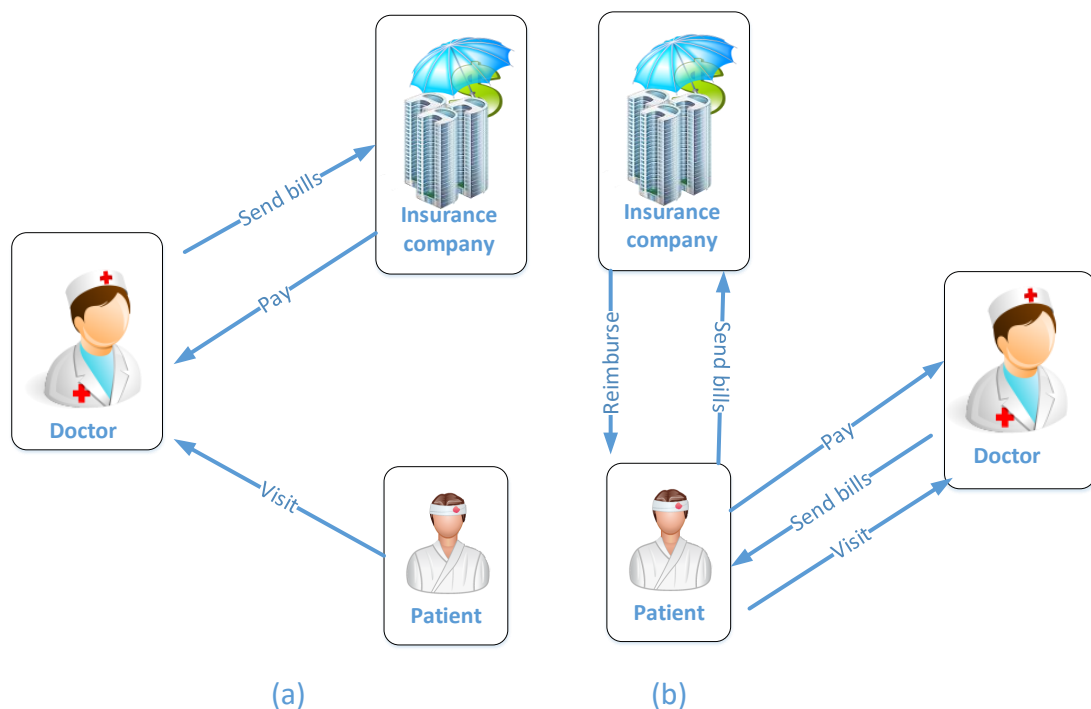


Figure 4: Two typical business models of billing

There is a different business model of billing as shown in Figure 4 (b). The doctor or the pharmacy asks the patient to pay the bill at first. Later the patient can send the bill to his/her insurance company to get reimbursed. We will discuss this model with more details in Section 6.2.3.2.

As we have addressed in Section 3.1.2, insurance companies involved in the billing process impose risk of disclosing the privacy of patients. In many existing eHealth systems, bills often include almost everything about patients' healthcare information. So insurance companies can build up database for storing the bills for their customers with patients' real identities. That imposes great threats to the privacy of patients due to the malicious behaviors from the corrupt insiders and the potential flaw of security

protection in the electronic systems of insurance companies. More details will be discussed in Section 6.2.

- Secondary Use

Secondary use of EHR is becoming more and more important and common. The ease of access, share, and manipulation of EHR facilitates the secondary use compared to paper based health records. In general, there are three important steps in secondary use of EHR as shown in Figure 5.

The first step is to “de-identify” (“anonymize” or “pseudonymize”) the health data. In Section 2.1 we already explained this step through some examples. The purpose of this step is to block the mapping from the health data for secondary use to the real identities of the participants. De-identification is often done by trusted third parties or the actual data creators (e.g. hospitals) in current eHealth systems. Therefore, patients usually have no idea about how their health data are “de-identified” or know the actual uses of their health data.

The second step is that secondary users get consent from patients. In current eHealth systems, patients usually are asked by a trust third party to agree or sign some disclaims which grant secondary users rights to use their data when patients enroll in eHealth systems. Patients are unable to control the uses of their health data afterwards. Some arguments about the necessity of patients’ clear consent exist in industry and research. Obtaining consent can be challenging and there have been major concerns about the negative impact of obtaining patients’ consent on the ability to conduct research. Such concerns are reinforced by the evidence that requiring explicit consent from participants in different forms of health research can have a negative impact on the process and outcomes of the research. For example, recruitment rates decline significantly when individuals are asked to consent; those who consent tend to be different from those who decline consent on a number of important demographic and socio-economic variables, hence potentially introducing bias in the results. Moreover, the consent requirements increase the cost of, and time for, conducting the research because of the very large population involved, the lack of contact channel between researchers and patients, and the time elapsed between data collection and the research study. Besides these negative arguments on getting consent, the procedure of getting clear consent from patients may itself disclose the patients’ identities to secondary users, because the consent may contain some information about the identities, e.g. name, date, location, and signature. Due to these issues, many current secondary uses don’t have the clear consent from patients.

The last step is that secondary users send feedback to the participating patients. In some cases, secondary users may have feedback (e.g. the statistic results or some participant-specific information which may interest patients) to the participants. For example, when a medical researcher on diabetes finds that a participant is prone to get diabetes based on the symptoms and diets of the patient, he would like to send a reminder to this patient; when a drug researcher finds a new cure scheme and new medicines for curing a patient’s illness, he may also want to notify the participants in time. The capability of transmitting feedback is a great incentive for patients to participate in the secondary use. However, in current eHealth systems, secondary users can

not contact patients due to the privacy issues, or the missing of contact information with the participants, the feedback is seldom sent out. For more details about the secondary use of EHR, please refer to Section 6.3.

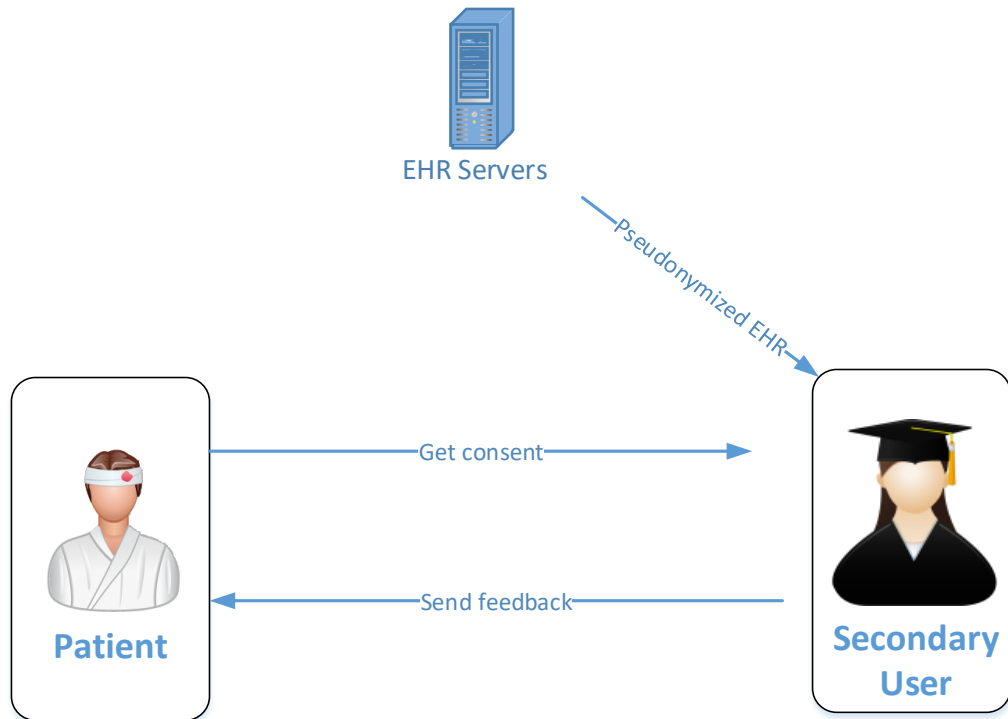


Figure 5: Process of secondary use of EHR

3.1.4 IT Infrastructure

- Storages

In a basic eHealth system as shown in Figure 1, the electronic health data are stored at different servers operated by different healthcare providers. For example, in Figure 1, Doctor A and Doctor B coming from two different hospitals provide healthcare service to the same patient, but they create and store EHR separately at Server A and Server B operated by two hospital independently; different pharmacies store the prescriptions and other information (e.g. sales records) into their own servers; insurance companies also save the billing information and all other information (like the customer registration) at their own servers.

As a result of dispersive storage, EHR of one patient are distributed divisively over the multiple healthcare providers (e.g. hospitals, practices) which the patient has visited as shown in Figure 6 (a).

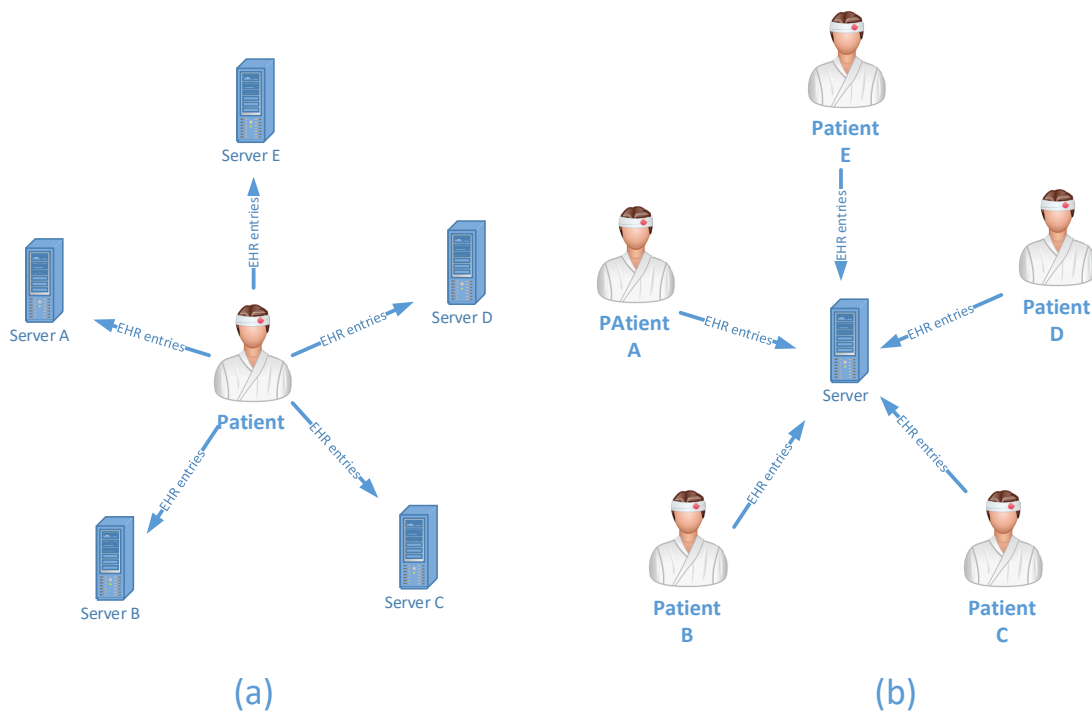


Figure 6: Two models of storing EHR

The dispersive storage model causes many problems (e.g. authorization, server reliability, different EHR formats) for those applications where all EHR of a patient need to be gathered together. E.g. a doctor who diagnoses a patient wants to review the anamnesis of the patient; a medical researcher collects all EHR of a certain kind of patients (Elkin et al., 2010). Especially for the secondary use of EHR, the scattered EHR in multiple locations will bring many other obstacles. One obstacle comes from the de-identification of EHR. As we have introduced in Section 3.1.2 and 3.1.3, because the real identity information of patients is often not necessary in the secondary use, the real identifiers need to be removed from EHR in advance (de-identification), in order to protect the privacy of participating patients during secondary use. It is, however, a technical challenge to remove patients' identity information consistently and unify the pseudonym of the same patient because EHR scattered in different locations are indexed by different identifiers (e.g. identity number, passport number, name and birthday). Another obstacle is that secondary users have difficulties in contacting patients to get consent for authorizing them to use EHR legally because they can not find the contact method of the participants from the de-identified EHR. Furthermore, secondary users will also encounter problems in sending feedback to the participating patients because the participants are not reachable.

An alternative way is to store EHR of all patients in a centralized server as shown in Figure 6 (b). EHR of one patient are indexed by a same identification and stored together. The sharing and gathering of EHR of patients will be much easier and more comprehensive than the dispersive manner. To this end, a cloud (Mell & Grance, 2011) is an ideal media for storing EHR, providing wide access to EHR, because it is able to

offer ubiquitous services to customers over the Internet (Rui & Ling, 2010). In the ordinary healthcare activities (e.g. patients visit doctors and pharmacies), all healthcare entities (e.g. healthcare providers, patients, health insurance companies) can access EHR from the cloud any time and any where, thus facilitating all these activities. E.g., doctors can view patients' previous health records stored in the cloud conveniently; patients can purchase medicines from pharmacies by using the electronic prescriptions stored in the cloud. Moreover, the cloud can provide great convenience for the secondary use of EHR due to the following reasons. Firstly, if EHR of a large population segment are stored in the cloud, secondary users have better chance to gather more extensive health information. Secondly, the computation power of the cloud can provide potential services to secondary users, e.g. the cloud can help a medical researcher to find target participants through powerful searching capability in all EHR stored in the cloud. Finally, because the cloud can perform as a media for the communication between secondary users and patients, secondary users are able to get the consent from the participants and send feedback to the latter through the cloud online more conveniently.

On the other hand, the public accessibility of the central server (e.g. cloud) brings the risk of disclosing patients' health information to ubiquitous attackers. The security of EHR and the privacy of patients will suffer from enormous threats when EHR are stored in a centralized manner. As a consequence, EHR must be encrypted and the encryption keys should only be known to the intended entities. In addition, the identity information of patients should also be strictly managed and never be linked to EHR by attackers. However, because patients' identity information needs to be frequently used in a typical eHealth system, the identities of patients confront high risk of disclosure. E.g., A doctor needs the identity information of patients to inquire their health insurance coverage from the health insurance companies; the cloud may need to check a patient's identity information in order to grant him to do any modifications on his EHR when the patient is managing his EHR online; secondary users may also need the identity information to confirm that the consent is exactly signed by the owners of EHR and the feedback are sent to the corresponding participants.

In this thesis, we adopt the centralized storage manner to store EHR due to the advantages introduced in the last paragraph (refer to Section 3.3 for more discussion on the centralized storage manner).

- Communication Network

The intranet is often used to share EHR inside one healthcare provider. For example, an internal network is deployed to connect the computers of all doctors and other staff in the same hospital. Patients' health data can be easily accessed inside the hospital for various uses. The intranet is an efficient communication approach inside a healthcare provider. EHR storage and communication can even be clear text which can facilitate the access speed and search. With the assumption that the staff inside are trustworthy, it provides high level of security and privacy because the potential outside attackers are unable to access the internal resources. However, there are several shortcomings by using the intranet to share EHR. One obvious point is that it is difficult to share EHR across different healthcare providers as we have introduced in Section 2.2.1.1. Another one is that the insiders (e.g. the administrators of the intranet)

are able to download all EHR of many patients, which will be a potential risk of disclosing the private data of patients. At last, for some healthcare providers, it is not easy for them to maintain such an intranet and servers for daily use due to the cost and technical reasons.

Some traditional ways such as telephone, fax and email are also used to communicate among different organizations in eHealth systems, e.g. healthcare providers may send the bills to insurance companies by fax. However, these traditional ways may easily be eavesdropped by attackers and malicious telecom service providers. Besides, they are also inefficient and inconvenient.

There are also some existing eHealth systems and research groups suggesting the Internet for communication. E.g., the Internet is used to share patients' EHR across healthcare providers; the Internet is used by healthcare providers to send the bills to insurance companies; patients can view and manage their own EHR through the Internet using any own devices. However, because the Internet is open (to the skillful attackers) and easily eavesdropped (e.g. eavesdropping at the bone routers), advanced security technologies must be used to protect the communication messages. In conclusion, compared to the traditional communication way, the Internet brings much easier connectivity and higher efficiency, but it also brings enormous challenges to design secure communication protocols to protect the transmitted messages.

3.2 Security and Privacy Threats

A practical eHealth system is a very complex information system, in which many roles, activities and transactions are included. Even in a small subset of the system (e.g. prescription system), the analysis on security and privacy threats is very complicated. Moreover, the structures of different eHealth systems are even different. As a result, instead of some standard threat analysis schemes (Schneier, 1999), we are going to analyze the security and privacy threats in some major aspects, which are commonly referred in different eHealth systems. In the following Sections, we consider the major roles and activities in eHealth systems, and analyze the common security threats to the EHR and privacy threats to patients. These analyses are not complete. However, these threats are common and important in different eHealth systems. Our solution in this thesis also mainly aim to resist these threats, and is capable to be used in these mentioned activities.

3.2.1 The Owner of Electronic Health Records (EHR)

Electronic health records contain highly personal information, from illnesses to family matters to emotional statuses. They have become invaluable collections of information used by a diverse group ranging from government agencies and disease researchers to marketing firms and for-profit data brokers (e.g. IMS¹¹)(Tanner, 2014). Government and for-profit businesses have long collected, parsed, and used collective patient data to track the path of chronic conditions and contagious diseases, follow the success rates of new and old treatments, develop new cures, and improve the quality of providers' services. However,

¹¹ Ims: <http://www.imshealth.com/portal/site/imshealth>

because the electronic health records are easily re-identified, and have different rules depending on state and organization, patients have little to no control over the information that reveals their very personal health information.

In many current eHealth systems, the actual holders of EHR are roles other than patients due to various reasons. For example, healthcare providers own and control EHR because they actually have created them and EHR are most often used inside healthcare providers. In some eHealth systems, the government or some trusted third parties are assigned to be the owners of EHR, and they are believed to protect the private information in EHR, and they are trusted to use EHR in a legal way.

Why don't many existing eHealth systems set patients as the owners of EHR? The most direct answer is that if patients own and control EHR, it will cause many inconveniences in many healthcare activities. For example, if many patients refuse the secondary use of their EHR, it will cause bias or error of the results. In medical research, the researchers need to use a large amount of de-identified patient data for developing new medicines and for finding relationships between disease factors and influences. Because it is known to many patients that the statistically de-identified data are not perfect (i.e., the de-identified data might be re-identified) (Bjurström and Singh, 2013), patients may have excessive worries but no incentive to contribute their private data for secondary use. These worries come true in some reports. E.g., in a study, the Whitehead Institute for Biomedical Research, a nonprofit research and teaching institution with programs in cancer research, developmental biology, genetics, and genomics, was able to re-identify 50 people who had provided personal DNA data for genomics studies. The odds of being named from a de-identified database were 4 in 10,000, according to a 2005 study¹². In the recent years, consumers share more identifiable information via social media and apps, and more information is digitally available, so perhaps they are more likely to be identified today.

We argue that patients should be the owners of their EHR. If a proper security and privacy solution is provided, the worries both from patients about their privacy and from other roles that use EHR can be dismissed. The solution should convince patients that they will not lose the control over their EHR, and the private information will not be disclosed in any case of uses. Moreover, the introduction of the security and privacy solution should not hinder the normal uses of EHR, e.g., EHR can be decrypted in time in the ordinary healthcare activities and secondary uses. In this thesis, we have proposed such a solution to set patients as the owners of EHR and enable patients to have full control in every use of their EHR (refer to chapter 6).

3.2.2 General Attackers

Because of the high concern on security and privacy, the health data are usually protectively (e.g. encrypted) stored in the servers to prevent the access of persons who are not authorized, and every use of the health data is very careful. However, these protected data have the risk to be attacked, even if the encryption itself is strong. According to the places where the attacker is, we divide the attackers into two types.

¹² From <https://dataflog.com/read/re-identifying-anonymous-people-with-big-data/228>

- With Ability to Access the Storage Servers

The first kind of attackers are the insiders in the systems as shown at the right of Figure 7. The insiders often refer to the malicious users who have rights to access the data stored at the server. Plain users like a nurse who can view a part of the data at the server are such potential insiders. It is even worse when an administrator of the server becomes a malicious insider. The administrator who has full ability to access the data server may completely destroy the privacy guarantee. Especially, because more and more healthcare providers outsource the IT work to some third companies, the data servers are physically out of the control of healthcare providers. In this case, the risk of insiders becomes higher.

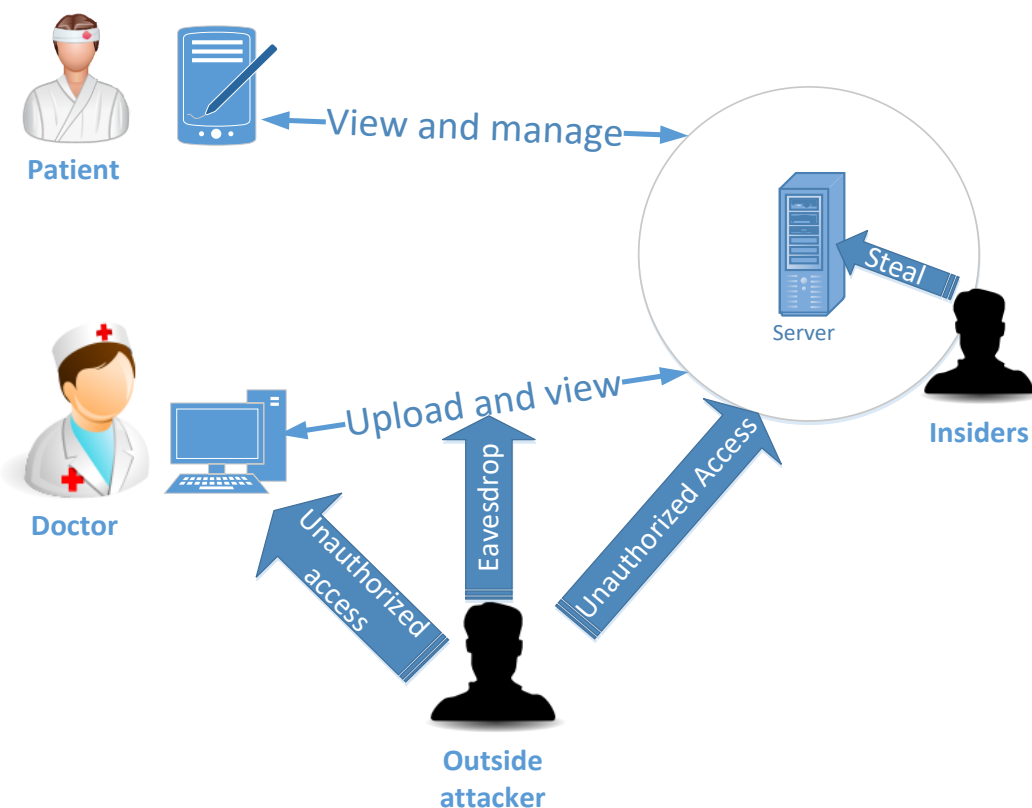


Figure 7: Examples for general attackers to EHR

Another scenario is that the security mechanism for protecting the data servers might be compromised by outside attackers through networks as shown at the bottom of Figure 7. Especially when the data server is connected to the Internet, the data server and the client computers of the doctors may suffer from high risk to be attacked. Although many advanced technologies are used to protect the IT infrastructure from invasion, the bugs or flaws caused by design and implementation still give chances to outsider. Depending on what information is obtained by the outsider, the data stored at the servers may be infringed to different degrees.

Moreover, some eHealth systems reside in public cloud (e.g. the eHealth in this thesis as introduced in chapter 6), the resources (e.g. EHR) are publicly available to anyone. Malicious users can download and analyze these data to discover interesting information, which may infringe the privacy of the data owners.

- With Ability to Eavesdrop the Network

When the users in an eHealth system access the data at the server through network, eavesdroppers have chances to obtain the private data. An example of such eavesdropping is shown at the bottom of Figure 7, where the communication between a doctor and the server can be obtained by an attacker. In fact, any other communications, e.g. between the patient's device and the server, have the risk to be eavesdropped. For the simplification, they are not drawn in Figure 7.

One kind of such eavesdroppers is the users who eavesdrop in the intranet of healthcare providers. Because some devices like intranet hub use broadcasting to transmit data, an eavesdropper who connected to the intranet can easily get the messages transferred among communicators and data server. In many cases, because the messages transferred over the intranet are not encrypted, it will cause serious privacy problems if a malicious eavesdropper exists.

If the messages are transferred over the Internet, some attackers who have ability to access the key devices (e.g. routers) in the bone network may eavesdrop and record the private data in the messages. This kind of attacks can be ruled out by encrypting the messages. However, the system designers must pay attention to the secret key management policy to avoid that the encryption key is leaked to the attackers. For example, a key exchange scheme is prone to be attacked by man-in-the-middle scheme (Seo and Sweeney, 1999), where an attacker impersonates as a transparent agent and obtain the encryption keys without being discovered by the communicators.

3.2.3 Healthcare Providers

In many existing eHealth systems, healthcare providers are assumed to be fully trustworthy. However, healthcare providers may still potentially disclose the privacy of patients.

One scenario is that the data stored in the servers of a healthcare provider may be leaked. A doctor in a practice knows the personal identities of all visited patients and all the private healthcare information. Although doctors rarely leak or sell the health data to someone else, we must take into account that the computer of the doctor may be compromised and the data stored in it might be obtained by the attackers. Some countermeasures will be discussed in our solution to reduce such kind of risk in Section 6.1 and 6.2. For example, patients can be anonymous to healthcare providers; EHR of patients are not stored on doctors' computers at all, or stored on doctors' computers encrypted by secret keys only known by patients.

Another scenario is that some healthcare providers may attempt to get illegal benefits from insurance companies. In many eHealth systems, a healthcare provider sends the bills to insurance companies covering patients to get paid. It is possible for a corrupt doctor (or an attacker who impersonate a doctor) to forge bills to get illegal benefits (Jesilow et al.,

1985). In our solution, we also propose some schemes to avoid this kind of behavior (refer to Section 6.2).

3.2.4 Insurance Companies

In many countries, people have health insurance covering the expenses of their health care served by healthcare providers (e.g. practitioners, hospitals, pharmacies). Especially in most European countries, the health insurance is mandatory to most residents, e.g. through some statutory insurance companies or public health fund. As we have introduced in Section 3.1.3, insurance companies receive and pay the bills from healthcare providers (where healthcare providers bill insurance companies directly on behalf of patients), or from patients (where patients receive the bills from healthcare providers and afterwards send the bills to insurance companies to get reimbursed).

The traditional bills usually contain all important information about the patients' illnesses, e.g., the operations and the medicines administered to patients. This private information will be completely disclosed to insurance companies who receive the bills. Insurance companies know whom the bills come from and which patients the bills originate from as the identities of both healthcare providers and patients are directly or indirectly enclosed in the bills.

One important reason why insurance companies hold such full information (both identities and EHR of their customers) is that insurance companies need to be able to detect fraud. In practice, some malicious patients who are not covered by health insurance or limitedly covered might steal the insurance information of other people to cheat insurance companies for invalid benefits. Hence, insurance companies have to firstly verify that the bills originate from the customers who are insured by them through checking the identities of their customers. Then, insurance companies also have to examine the details of the operations and medicines to find out whether or not all the expenses in the bills are in the range of the insurance coverage of the corresponding patients. Another fraud example comes from the corrupt doctors or pharmacies who may forge bills to get invalid benefit as discussed in Section 3.2.3. The second main reason is that insurance companies also claim that they need to conduct some statistical analysis on the healthcare information for the benefits of patients (e.g., in the assessment of new treatment approach or medicines, in the evaluation of healthcare providers).

In many current eHealth systems, patients' private health information with their real identities is fully disclosed to insurance companies due to such reasons. Although insurance companies declare that they will guarantee the privacy of patients under the restraining of laws, patients still doubt that insurance companies might use their health information for some present or future purposes without their consent even with their identities disclosed. Moreover, since the health information of patients is stored in the servers operated by insurance companies, there exists risk that the data may be stolen by malicious insiders or skilled attackers through compromising these data servers.

3.2.5 Secondary Users

- Re-identify the Participants from De-identified EHR

The de-identified EHR provided to secondary users sometimes might be re-identified as we have introduced in Section 3.2.1. Because the anonymized or pseudonymized EHR usually keep some insensitive identifiers (e.g. gender, age and postcode) as introduced in Section 2.1.3, patients' real identities may be found out (re-identified) from EHR through some advanced information analysis technologies as mentioned in Section 3.2.1. Especially in some worst cases, the de-identified health data are owned and maintained by some third organizations that sell or freely provide EHR to any interested people. That will bring great risk to the privacy of patients due to the various purposes for which different people obtain EHR.

Table 4: An example to re-identify the patient in secondary use

No.	Name	Post code	Gender	Age	Height	Data
1	Pid ₁	53127	Male	68	175	...
2	Pid ₂	*	Male	68	175	...
3	Pid ₃	53127	Female	60	170	...
4	Pid ₁	*	*	68	175	...

Taking the example from Section 2.1.3 as shown in Table 4 which are the pseudonymized EHR obtained by a researcher, we explain how to potentially re-identify a patient from the de-identified EHR. If the researcher lives in the area with postcode 53127, and knows most people there, he might find that there is only one man of the age of 68 with height of 175. Obviously, he can know the first row in Table 4 comes from the person that he knows (e.g. name). Then from that row he also learns that his pseudonym is Pid₁. Consequently, he may at least find out all the rows with pseudonym Pid₁ in the database (e.g. the line 4), and discover almost all the health information of this patient.

- Find out Patients' Identities in Inquiring Consent

Although in many current eHealth systems, patients' health data are used for secondary use beyond the full control of patients, we argue that patients are the real owners of the health data and therefore secondary users must get clear consent from patients as discussed in 3.2.1. However, the process of inquiring consent may leak the identity information (e.g. contact method) of patients to secondary users or any organizations that are in charge of consent management. Moreover, because the consent has to include signatures or any other forms of verification signs, the consent is traceable and can potentially be linked to the real identities of patients. That is to say, patients' full ownership and control over the health data, which aim at protecting the privacy of patients, unexpectedly conflict with the privacy of patients because of the necessary communication in the consent management. We need methods to solve the conflict.

- Disclose the Privacy of Patients in Sending Feedback

When a secondary user sends personal feedback to a patient as introduced in Section 3.1.3, the privacy of the patient and identity may be also disclosed as shown in Figure 8 through the following ways. Firstly, the feedback containing the private information about patient’s health status may be disclosed. If the feedback is obtained and read by some people other than the intended patient (through attacking the computers of the patient and the secondary user or eavesdropping the network), it will cause privacy problems. Secondly, when the secondary user sends feedback, some identifier of patients must to be leaked to the secondary user or the mediate organization that transfers the messages. For example, the secondary user or the mediate organization knows the receivers’ contact methods like email or address, which can be used to deduce accurate identity information about the patient. At last, the feedback sending through email or other communication methods may also potentially leak the private information to the IT service providers (like email service companies). Due to these threats, there must be a security approach to protect the messages transmitted over the network, and patients’ private identifiers (like contact method) should be kept secret to any unnecessary persons (like secondary users and mediate people).

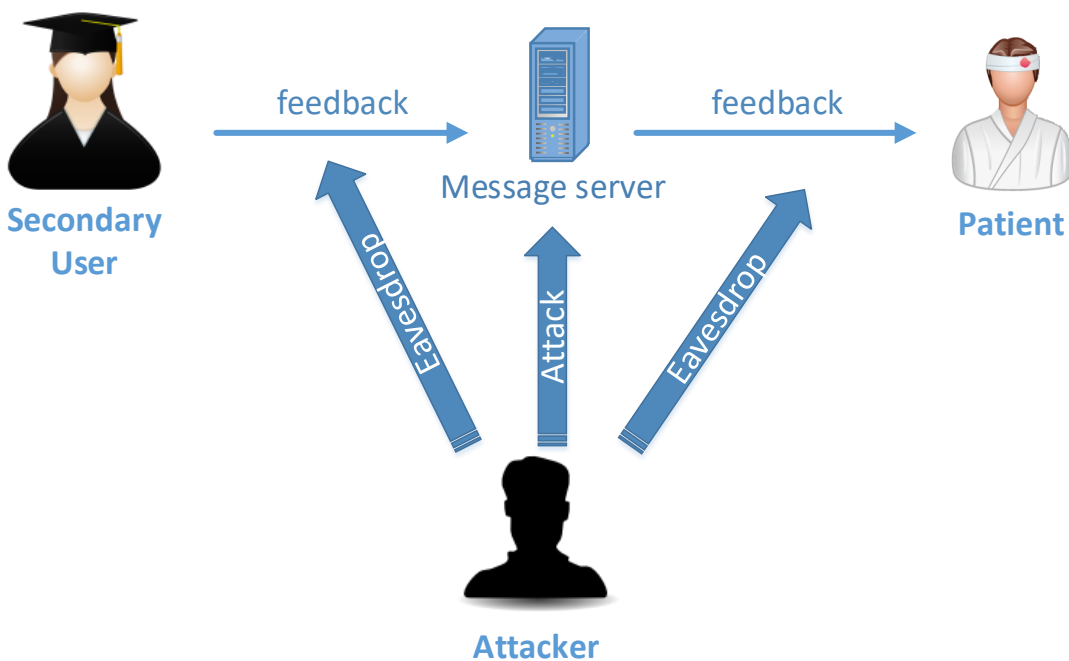


Figure 8: Privacy disclosure when a secondary user sends feedback to a patient

3.2.6 Trusted Third Parties

- Insiders in Trusted Third Parties

It is risky to use any trusted parties with powerful rights (e.g. those in charge of performing secret key management and storing the secret keys at their servers) in eHealth systems as we already discussed in Section 3.1.2. A malicious insider can

easily get the secret keys of patients and read patients' private information. More seriously, an insider can even impersonate patients with the secret keys to get illegal benefits. Even worse, these insiders are difficult to be discovered as the secret keys are sometimes the critical and unique credentials in many transactions.

Although the semi-trusted third parties (as introduced in Section 3.1.2) are unable to behave as fully trusted third parties to discover the privacy of patients directly, they can also do something evil. A semi-trusted third party (e.g. a certificate authority) can easily create a non-existing person, and use the created person to get illegal benefit from the service providers in eHealth systems if the validations are only executed at the semi-trusted party.

- **Compromised Trusted Third Parties**

Besides the insiders, we must also notice the risk of that the trusted third parties might be compromised by attackers. Because the trusted third parties usually possess many important data in their servers, they are often the intended targets for the attackers who already have some parts of patients' information (e.g. the attackers who have obtained some EHR by eavesdropping). Especially some trusted third parties also provide online service like real-time authentication. They may confront high risk of being attacked. The compromised trusted third parties will bring catastrophic influence to the privacy of patients and even the whole eHealth systems.

In the design of eHealth systems, the threats from the trusted parties must be carefully considered, and the power of them should be decreased as much as possible. A case where TTP with limited power will be introduced in Section 6.1.1, where we only use Certificate Authority (a kind of semi-trusted TTP) to issue certificates to healthcare providers other than the patients.

3.3 A Cloud-based Model of eHealth Systems

3.3.1 Principles for the Cloud-based eHealth Model

3.3.1.1 Rule out the Fully Trusted Third Parties

Basing on the survey into the current eHealth systems in industry and academic research, we propose a cloud-based eHealth system model as shown in Figure 9. In this model, we exclude all unnecessary trusted third parties which may cause risk of leaking the private information of patients. As we have discussed in Section 3.1.2 and 3.2.6, because a corrupt insider may appear inside a trusted third party or a skilled outside attacker may compromise the servers at the trusted third party, an eHealth system, where patients' highly private information is stored, getting rid of the risk of disclosing the privacy of patients due to such fully trusted third parties is much preferable.

Instead, we introduce in our model an entity named "certificate authority". It issues digital certificates to the specified entities. Doctors, pharmacists, insurance companies and secondary users need to apply for certificates from the certificate authority. Generally speaking, it is a normal process in many public key infrastructure (PKI) based systems. Each certifi-

cate applier needs to generate a self-known private key and a corresponding publicly known public key. The public key together with some necessary information of the applier (e.g. name, certificate type and valid period) is signed by the certificate authority to form a digital file, i.e. the applier's certificate (Tuecke et al., 2004).

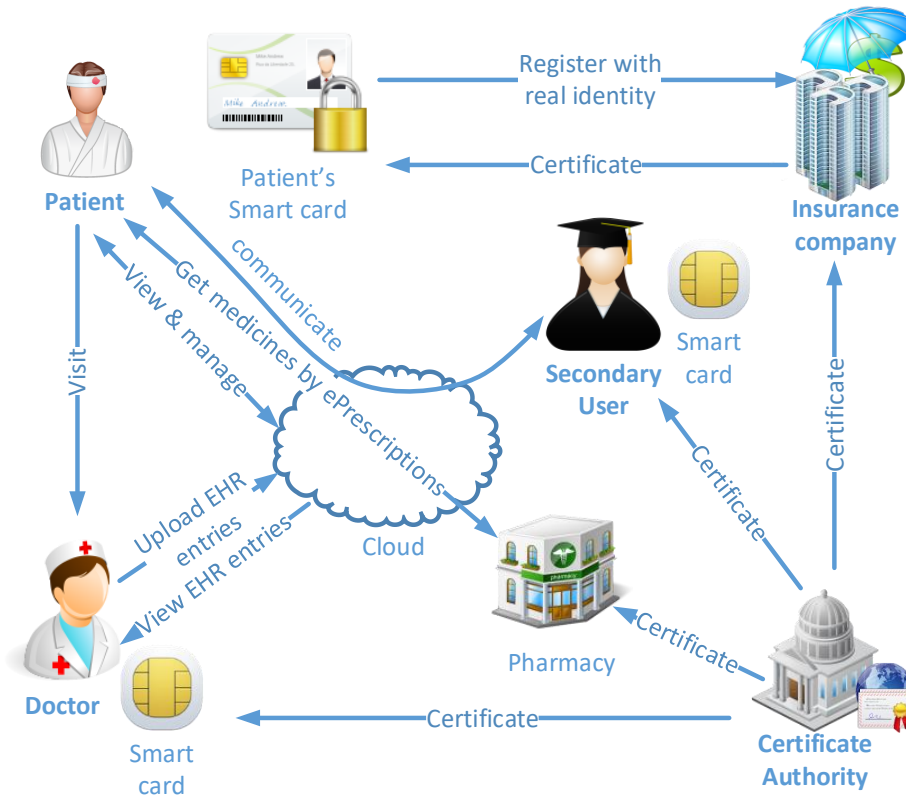


Figure 9: An eHealth model for ordinary healthcare activities and secondary use

The certificate authority is one kind of semi-trusted third party which ensures the relying on the signatures or other assertions made by the entities in the system. It brings very limited threats to the security of patients' EHR and private information of patients as we introduced in Section 3.1.2. Because the private keys in our model are only known by the certificate appliers, any corrupt insiders or outside attackers are unable to disclose the encrypted content or forge the signatures from knowing the public keys. In practice, the certificate authority can be the health department of the government or an international trusted organization which checks the qualifications of doctors, pharmacists, insurance companies and secondary users, and issues them digital certificates. The certificates and the corresponding private keys of the appliers are stored and protected by the smart cards owned by the appliers with authenticating PINs to avoid abuse of the smart cards (refer to the next Section). A corrupt certificate authority might issue invalid certificates to an unqualified person (e.g. a fake doctor, potential attacker). However, the unqualified person is not able to get a patient's private information unless the patient visits the unqualified person by mistake. In other words, the corrupt or compromised certificate authority can only cause limited affection to the private data of the patients.

3.3.1.2 Utilize Smart Card for Trust Computing Module

We render the smart cards of a doctor and a secondary user as examples in Figure 9, besides the patient's. Actually, all other roles (e.g. pharmacies, insurance companies) can possess smart cards to store their secret keys, digital certificates and any other information in reasonable size. These smart cards are not depicted in the Figure 9 for simplicity.

A smart card is typically a pocket-sized card (or any other shapes) with embedded integrated circuits. There are contact smart cards and also contactless ones. The independent (not relying the connected devices except the power supply) hardware and software in the smart card are able to execute some light-weight programs such as encrypting/decrypting and digitally signing with the respective secret keys. The secret keys are stored in the protected memory of the smart cards. It is difficult to read the secret keys directly from the protected memory thanks to hardware protection. Due to the portability in size and the security functions, they are ideal for storing the secret keys in eHealth systems. It is convenient and multifunctional for all entities to take along the cards when they are participating in all kinds of healthcare activities.

The use of smart card is now a common approach for the users to carry secret keys (Rankl and Effing, 2010). Many current eHealth systems are adopting smart card for identification, authentication, billing and so on. Due to its high resistance to attacks, smart card can act as the trust computing module for the security and privacy in an eHealth system.

3.3.1.3 Adopt Cloud as Storage and Communication Media

As we already discussed in Section 3.1.4, we recommend the central storage method to store EHR of patients. Cloud, especially the public cloud, is obviously the ideal technology for such a central storage due to the following reason. Firstly, cloud can provide convenient sharing of the data among all entities in eHealth systems, which is exactly needed for sharing patients' EHR among different healthcare providers. Secondly, a cloud can decrease the cost and the technical difficulty of maintaining servers by individual healthcare providers. Thirdly the cloud can provide powerful additional services, like searching in the stored EHR. Finally, the cloud can also be an ideal communication media across different roles. E.g., the communication between secondary users and patients for consent management and feedback dispatch can be carried out over the cloud without another third message transmitter.

3.3.1.4 Protect EHR and Messages for Security and Privacy

Because EHR, in which patients' highly private information is stored, are transmitted over the publicly accessible cloud, the security of EHR and the privacy of patients should be carefully considered. For example, in our solution, we suggest to use pseudonyms to protect the identities of patients. The pseudonyms are generated by patients from the secret keys only known by patients. They will be used to index EHR contents stored in the cloud. Moreover, the pseudonyms are directly and indirectly related with security and privacy functions. For more details about the design and application of the pseudonym solution, please refer to chapter 5 and 6.

3.3.2 Refined Procedures and New Threats

3.3.2.1 Registration at Insurance Company

The process for each patient registering at an insurance company remains almost the same as described in Section 3.1.3. Specially, each patient will generate a pair of private key and public key, and the private key is only known by the patient. During the registration, the patient provides the public key to the insurance company. The public key combined with some other information (e.g., the registration reference number and the identifier of the insurance company) will be signed by the private key of the insurance company to form a digital certificate of the patient. Both the public key and the certificate from the insurance company will be stored in the patient's smart card. The certificate will be publicly accessible, while the corresponding private key must be stored in the protected memory of the smart card.

One new threat is that the insurance company can also access EHR in the cloud. Under the setting that the insurance company is not allowed to know the details of patients' health information, the insurance company is able to combine the information that it has mastered (e.g. registration and bills) and the public available data in the cloud to discover all the private health information of patients. There must be some protection scheme to prevent the insurance company from the linking. For example, in our solution we utilize a novel pseudonym scheme and corresponding signature scheme for the billing process. Technically, they protect the identities of patients in the bills against insurance companies, and at the same time, insurance companies are able to validate that the bills originate from the customers insured by them. Our solution excludes any fully trusted third party that could impersonate patients or discover all the private information (e.g. disease history) of patients in the billing procedure. For more details, please refer to Section 6.2.

3.3.2.2 Visit to Doctor

A patient takes along the smart card in which the patient's information (e.g. certificate and private key, secret key, and photo) was embedded to visit a doctor. The doctor can validate the patient by challenging the patient's private key and check the certificate in the smart card with the help of the insurance company (e.g. CA service) (Tuecke et al., 2004). After the doctor created a record (including examinations, diagnosis, curing, prescription and so on), the patient generates an encryption key based on the patient's secret key to encrypt the record. Then the encrypted record is uploaded to the cloud to form an EHR entry of the patient's whole EHR. The encryption of record is executed in the patient's smart card, i.e., the encryption key is even not known by the doctor.

If the doctor wants to view a patient's anamnesis, the EHR entries previously stored in the cloud must be downloaded on the network. The downloaded EHR entries then need to be decrypted by the encryption keys regenerated by the patient's smart card. The decryption of EHR is also executed in the patient's smart card.

Some new security and privacy issues related with cloud will appear. For example, the doctor has to be authenticated by the cloud to avoid illegal uploading of the patient's EHR. What's more, when the doctor downloads EHR entries of the patient, the cloud may find out that the downloaded EHR entries from one source come from one patient, which might help the cloud to discover the identity of the patient.

Moreover, if the patient wants to manage or update the information in one EHR entry uploaded to the cloud afterwards, the cloud also has to authenticate that the patient is indeed the real owner of the EHR entry. In the authentication procedure, the patient might be identified and leak the privacy to the cloud.

3.3.2.3 Medicine Purchase at Pharmacy

With the help of the cloud, a patient does not need to take any paper document (e.g. prescription) from a doctor to get medicines at a pharmacy. Because the prescription was stored in the cloud too as introduced in Section 3.3.2.2, the pharmacy can download it from the cloud and sell the medicines to the patient after some simple validations.

Because of the public access to the prescription in the cloud, it is very important for the pharmacy to check the validity of the prescription and that the prescription belongs to the patient to avoid abuse of the prescription (e.g., another patient may illegally use the prescription to purchase medicines from a pharmacy). We will discuss more details of such security threat and countermeasure in Section 6.1.

3.3.2.4 Billing

We mainly consider the first kind of billing model introduced in Section 3.1.3. With the help of the cloud, the bills from doctors and pharmacies can be stored in the cloud. Then later the cloud pushes the bills to insurance companies. We don't depict the billing procedure in Figure 9 for simplicity.

The bills stored in the cloud introduce new risk of disclosing the content in the bills to attackers who can access the data in the cloud. What is more, forged or modified bills may be easily created for illegal benefits by malicious attackers. The security of bills should be re-considered. More details of the new billing procedure and the security issues in a cloud-based eHealth system will be introduced in Section 6.2.

3.3.2.5 Secondary Use

Because EHR of patients are stored in the cloud, secondary users are able to conveniently obtain the health data of a large population from the cloud. For example, a medical researcher wants to get EHR for all patients with diabetes, the researcher just need to surf on the cloud, and search for all EHR entries including key words "diabetes" or "insulin". The cloud can provide these matching EHR entries to the secondary user directly after the authentication on the identity of the researcher (each secondary user has a valid certificate as introduced in Section 3.3.1.1).

However, because EHR are encrypted by patients' encryption keys which are not known by the cloud or secondary users, secondary users are unable to decrypt them. Moreover, it is still a problem when secondary users communicate with patients to get consent and send feedback.

In this thesis, we utilize a pseudonym solution to protect the privacy and guarantee the full control of patients during the consent management. This dismisses the patients' worries on the secondary uses of their private health data. We also provide a novel approach to enable

secondary users to send feedback to patients without disclosing any private information. This improves the incentives of patients to contribute their EHR for secondary uses. For more details on the challenges and countermeasure in our solutions, please refer to Section 6.3.

4 State of the Art

4.1 Pseudonym Schemes for General Purpose

Although many familiar systems are not clearly claimed as pseudonym systems, they are some kinds of pseudonymous systems. For example, many instant communication systems (e.g. Skype¹³, telecom systems) generate and assign virtual “id” for their users, or the users can register and set their own personal usernames (e.g. some public forums do not require the real identities of the registered users). The “id” or the “usernames” can be considered as the users’ pseudonyms. These pseudonyms can be linked to the real identifiers (if provided in the registration by the users) only by the service providers. The central service providers perform as trusted parties to manage these pseudonyms and be in charge of protecting the privacy of the users.

Generally speaking, a pseudonym scheme can be easily designed and deployed if a powerful central party exists. An organization (e.g. the central service provider in the above paragraph) that is considered as a central party can use many cryptographic algorithms (A) to generate pseudonyms (PID) for users (U) to fulfill the requirements listed in Section 2.1.1. The organization usually keeps a secret key (sk) to produce the pseudonyms and manage the users. For example, a typical algorithm in A is a keyed hash function (Krawczyk et al., 1997). The following describes how the organization uses a keyed hash function $KHash_{sk}$ to generate a pseudonym (PID_i) for each user U_i . $PID_i = KHash_{sk}(id_{u_i})$ where id_{u_i} is an identifier (e.g., name and identity number) of the user U_i , and $KHash$ can adopt many cryptographic hash functions (Rogaway and Shrimpton, 2004). There are several benefits to generate the pseudonyms by a central party. Firstly, this kind of pseudonym systems with powerful central organizations are easy to deploy due to the simple structure and algorithms used. Secondly, because all pseudonyms are generated by a unique organization, it is easy to guarantee that the pseudonyms are collision free. What is more, because the central organization is also in charge of the management of pseudonyms, users’ pseudonyms can easily be maintained, e.g. created, tracked and revoked.

However, this kind of central party based pseudonym schemes have many limitations when they are applied to complex systems. The first limitation is that the pseudonyms can only be used inside one organization. When there exist many organizations in a system and the users need to move from one organization to another, these simple pseudonym schemes can not provide expected privacy protection. E.g., the users’ behaviors can be easily traced through the transaction data in many organizations because each user has only one unique pseudonym. This will be a big threat to the users’ privacy if malicious insiders exist. For an outside attacker, although the traces of users are apparently anonymous (because the outside attacker does not know the identities of the users directly), it is still risky for disclosing the real identities of the users if some auxiliary information is available and some advanced information analysis technologies are used, as introduced in Section 3.3.2.5. The second limitation is that the powerful central organization may disclose the identity information of

¹³ <http://www.skype.com>

the users. Because the central organization with secret key (sk) is able to reversely link the pseudonyms to the real identities of the users, insiders in the organization are one major risk to disclose the users' identities intentionally or unintentionally. In addition, the central organizations may be compromised by the skilled attackers. As a result, the private information of the users is seriously threatened.

The modern pseudonym systems were firstly introduced in (Chaum, 1985; Chaum & Evertse, 1987) and later with more work (Chaum, 1995, Chaum and Pedersen, 1993). They proposed a way for users of information systems to avoid being traced in different transactions by using pseudonyms which are not linkable to users' real identities. They also provided an implementation by RSA. Each user owns different pseudonyms at different organizations instead of using a unique one. Thus the organizations can not collude to trace the users. Moreover, each user can use the credentials issued by organization A at organization B without disclosing the user's pseudonym to organization A. However, their schemes rely heavily on the trustworthiness of a trusted third party (TTP). The TTP is in charge of pseudonym generation, authentication and even the validation of credentials. This imposes risk of leaking the privacy of the users similar to central parties in the last paragraph.

Soon after that, some researchers proposed other pseudonym schemes trying to release the high dependency on the TTP. In (Damgård, 1990), the author presented a new pseudonym model, where TTP is only in charge of issuing the pseudonyms to the users (the generation of pseudonyms). However, this scheme was based on zero knowledge proof, which makes this scheme ineffective in practice. Later, in (Chen, 1996), a more effective implementation of Damgård's model based on discrete logarithm was presented. However, in this scheme TTP is also required to be in charge of the pseudonym authentication besides the pseudonym generation.

In (Lysyanskaya et al., 2000), a pseudonym scheme was proposed to further decrease the power of TTP in pseudonym systems. In this scheme, they define a user clearly as "an entity with a secret key", and each user chooses a secret key by himself/herself. The secret key is only known by the user. Moreover, the pseudonyms are generated by users' secret keys and organizations' secret keys together. No single powerful TTP is needed in the system. Instead, the organizations and users cooperate to generate, authenticate the pseudonyms and validate the credentials. This scheme rules out the requirement of TTP in the system, and disperses the TTP's functions to the organizations. The attackers can only hack a single organization to get limited private information about the users.

Except these general pseudonym schemes, there also exist many special pseudonym schemes designed for specific systems. For example, in email systems (Mazieres and Kaashoek, 1998), RFID (Molnar et al., 2006), and VNET (Lu et al., 2012). These pseudonym schemes aim at different application scenarios and requirements. However, eHealth systems are much more complex in structure and include much more entities and transactions. It is not easy or unclear to port these pseudonyms designed for those specific systems to eHealth systems.

4.2 Pseudonym Schemes for eHealth Systems

Pseudonym schemes are widely adopted in the secondary use (e.g. medical research) of health data. When patients' health data are reused for the purposes (e.g., for medical research) other than ordinary health care, because the real identities appeared in the data are not necessary and for the sake of protecting the privacy of patients, the real identities of patients need to be removed (often annotated as "de-identified") as we have introduced in Section 3.2.5. Because of the potential advantages brought by pseudonyms as discussed in Section 2.1.3, the de-identification process is usually carried out by a trusted third party (TTP) who replaces patients' real identities by pseudonyms in current eHealth systems. In (Noumeir et al., 2007), hash functions are mainly used by a TTP to compute the pseudonyms from the identifiers (e.g. name). In (Pommerening and Reng, 2004), a TTP uses a secret key to encrypt patients' identifiers to generate their pseudonyms. In both pseudonym schemes, the TTP has the ability to reverse the pseudonyms to patients' identities by decrypting with the secret key or by a hash table, which can facilitate the consent management and communication between secondary users and the participating patients as introduced in Section 3.1.3.

However, the existence of a TTP is a weakness on security and privacy in the system as we have discussed in Section 3.3.1.1. Many pseudonym schemes were proposed to weaken or rule out the TTP. In (Iacono, 2007), a complex pseudonym scheme based on hash and RSA was presented. In this scheme, patients' real identities are not directly known by the TTP, but maintained by the trusted local pseudonym centers. Thus the risk of disclosing patients' identities due to a single corrupt or compromised TTP decreases slightly. In (Elger et al., 2010), they proposed a pseudonym scheme using both block cipher AES and Hash functions to generate reversible pseudonyms for patients' health data from different hospitals. The pseudonyms are generated from the secret keys known by the hospitals that own (or create) the health data instead of any other trusted parties. Also the hospitals are in charge of reversing the pseudonyms to patients' identities by their secret keys. In (Lehnhardt and Spalka, 2011), a pseudonym scheme without a single TTP was proposed, and it is based on the cryptographic algorithm ECC. This scheme only requires a central server whose responsibility is to control the users' pseudonyms in order to avoid collision of pseudonyms. The pseudonyms of patients are generated by the patient-only-known secret keys. In (Dubovitskaya et al., 2015), the pseudonyms are generated based on multi-key searchable encryption by the secret keys both from patients and doctors. The trust base of the privacy of patients turns to doctors.

Besides the pseudonym schemes proposed in secondary use of health data, there are also some researchers focusing on the pseudonym schemes that can be used in ordinary healthcare activities (e.g. visiting doctors).

In (Neubauer and Heurix, 2011, Riedl et al., 2008), public key cryptography is used to generate patients' pseudonyms, which index patients' EHR at health sever. In these pseudonym schemes, a pseudonym server which is in charge of the pseudonym generation and lookup is required. The pseudonym server should be fully trustworthy, which imposes risk to the privacy of patients.

In (Zhang and Liu, 2010), the pseudonyms of patients are generated from the unique identity number by a hash computation. The pseudonyms are computed when EHR are created

and they are the root nodes of trees to index EHR. Patients and doctors can use the pseudonyms to search and share EHR in the future healthcare activities.

In (Benzschawel and Da Silveira, 2011), they proposed an eHealth system with pseudonymized EHR stored in multiple servers according to data type. Patients' pseudonyms and identities are centrally managed by a powerful trusted third party. However, the pseudonym generation algorithm was not specified in the paper.

In (Li et al., 2011), a complex eHealth cloud was proposed to store EHR online based on pseudonyms and encryptions. Patients' pseudonyms are generated by doctors when patients visit doctors. Each patient owns a patient-only-known key seed (KS). KS is used to generate a new pseudonym through a hash computation on serial number (SN) and a random number (R) by the doctor each time the patient visits the doctor. SN and R are generated by the doctor, and they are uploaded separately to two trusted servers. Meanwhile, KS, SN and R are used to produce an encryption key (to encrypt the an EHR entry) through another hash computation. The patient's pseudonym together with the encrypted EHR entry are uploaded to a health cloud. When another doctor wants to retrieve the patient's EHR afterwards, the doctor firstly has to fetch the SN and R from the two trusted servers, and with the patient's KS, to reproduce the pseudonym and the encryption key. Then the pseudonymized and encrypted EHR stored at the health cloud can be downloaded. In this pseudonym solution, the cloud is not necessary to be trustworthy because of the pseudonymization and encryption of EHR. However, although the two separate trusted servers used decrease the risk from insiders or attackers at a single TTP, it does not eliminate the powerful TTP all together. Moreover, the system relies on two online TTP servers, which may potentially cause problems of availability and delay.

4.3 Comparison of Existing Pseudonym Solutions

Because the pseudonym schemes reviewed above aim at different purposes in eHealth systems, we generally list the features of these previous solutions in the several aspects as shown in Table 5 to compare their advantages and shortcomings.

4.4 Expected Revolutions in Pseudonym Schemes for eHealth Systems

- Restore the Ownership of EHR to Patients

As we have discussed in Section 3.2.1, patients should be the real owners of the health data. No matter what kinds of pseudonym or encryption schemes are used for protecting the security and privacy, patients' acknowledgment and consent for any use of their health data must be guaranteed. Patients should be able to control and manage their private information. In most current eHealth systems and research work, this principle is not properly followed. Instead, healthcare providers (like doctors, hospitals) actually "own" the data and have the full privilege to use the health data of patients for any purpose under unclear consent and insecure privacy protection.

When a pseudonym scheme is used in an eHealth system, patients should have full control on their pseudonyms, instead of that other parties (e.g. healthcare providers, TTP) generate and manage the pseudonyms.

Table 5: A comparison of existing pseudonym solutions in eHealth systems

Pseudonym solutions	Cryptographic base	TTP used	Application purpose	Cloud involved	Notes
Noumeir, Lemay, & Lina, 2007	Hash function	Powerful, central	Secondary use	no	Collision considered
Pommerening & Reng, 2004	Block ciphers	Powerful, central	Secondary use	no	Easy to reverse the pseudonyms
Iacono, 2007	RSA	Powerful, less central	Secondary use	no	Use local clinics as TTP
Elger et al., 2010	AES, hash	Powerful, less central	Secondary use	no	Hospitals owns secret keys to reverse the PID
Lehnhardt & Spalka, 2011	ECC	No TTP, but a central control server	Secondary use	no	Pseudonyms are collision free
(Dubovitskaya, Urovi, Vasirani, Aberer, & Schumacher, 2015	multi-key searchable encryption	No TTP, trusted doctors	Secondary use	no	Patients and doctors use their secret keys to generate PID
(Neubauer & Heurix, 2011; Riedl, Grascher, Fenz, & Neubauer, 2008	PKI	Powerful	Ordinary health care	no	
Zhang & Liu, 2010	Hash function	No TTP	Ordinary health care	no	Hash tree used for fast search
Benzschawel & Da Silveira, 2011	-	Powerful	Ordinary health care	no	Use distributed servers to lower risk
Li, Chang, Huang, & Lai, 2011	Hash, block cipher	Powerful	Ordinary health care	yes	Two separate TTP to decrease risk
This thesis(Xu and Cremers, 2014a, Xu and Cremers, 2014b, Xu et al., 2014)	Hash, DLP	No TTP	Whole eHealth systems	yes	Refer to Chapter 5~7

- Lower the Power of Trusted Third Party

As we have discussed in Section 3.2.6 and 4.2, the existence of a TTP brings potential risk of disclosing the privacy of patients from many aspects. However, many current eHealth systems and research work have granted unlimited power to the TTP which knows too much private information about patients. The consequence of corrupt or compromised TTP must be seriously estimated in advance if a TTP is used in the system.

Many existing pseudonym schemes use a TTP, which plays an important role in generating, managing and/or authenticating pseudonyms. Sometimes, it is not possible to rule out the existence of a TTP, but we should lower down the power (the ability to disclose the users' privacy) of the TTP as much as possible.

- Protect the Privacy of Patients Everywhere

Most existing pseudonym schemes only work in some chosen healthcare activities as introduced in Section 4.2. However, the privacy of patients is equally important in every process. Using different security schemes in different processes not only increases the complexity of the system, but also brings potential risk of disclosing the privacy of patients due to one single flaw in one security scheme. It is desirable that a uniform security scheme is used for protecting the privacy of patients everywhere with strong and provable security level.

- Use Cloud to Serve eHealth Systems

The cloud is becoming more and more important and popular in our real life. With the help of a cloud, eHealth systems will provide more convenient and effective healthcare service as we have introduced in Section 3.1.4. In some current eHealth systems and research work, the cloud is more and more involved in serving as the storage media of EHR and even performing as a TTP. However, the introduction of a cloud also brings great challenges to the security and privacy as we have discussed in Section 3.1.4. There must be an appropriate solution to deal with the challenges while keeping the advantages that a cloud brings into play.

A pseudonym scheme applied in a cloud environment must face much more challenges. The attacks may come from anybody who can access the cloud. A pseudonym scheme must consider the ubiquitous threats, besides it should be practical to be used in such an eHealth model with cloud as introduced in Section 3.3.

5 A Novel Pseudonym Solution

5.1 Features of the Pseudonym Solution

- Basic Requirements

We have discussed some basic requirements (e.g., collision-free, one-way, independence) to a pseudonym scheme in Section 2.1.2. In the following we introduce further some basic requirements to the pseudonym solution.

Size and computation. The size of the pseudonyms should be reasonable (e.g. 256 bits) to save storage space and decrease the network load. The generating of pseudonyms should be efficient in current hardware and software condition.

Multiple pseudonyms. Each user can have multiple pseudonyms. These pseudonyms should be easily generated and reproduced from knowing the secret key.

Encryption with pseudonyms. Each pseudonym must have a corresponding encryption key, which will be used to encrypt the data responding to the pseudonym.

One-way. Each user's pseudonyms are computed from the user's identifiers and secret key. It should be difficult to compute the identifiers and secret key from the user's pseudonyms.

Collision-free. Each user has multiple pseudonyms and a pseudonym system has many users. Two arbitrary pseudonyms no matter from the same user or two different users must have trivial probability or impossibility to collide.

Independence. An attacker is not able to find out which pseudonyms belong to one user. Even if the attacker knows one or more pseudonyms from one user, he/she can not deduce other pseudonyms of the user.

- Decentral without TTP

There is no trusted third party (TTP) involved in our pseudonym solution. So the users' pseudonyms are not generated by any TTP or the service providers, but generated by the users themselves. The users will generate their pseudonyms from MSK introduced in Section 5.2. Because there is no such a central server for maintaining the pseudonyms, the collision control of the pseudonyms will face new challenge.

- Authentication in Cloud Environment

The authentication of pseudonyms as introduced in Section 2.1.2 will induce new problems in the cloud environment. Due to the distribution feature of the cloud especially the public cloud, the cloud can not be considered as a trustworthy service provider that can be expected to keep the private information of the users any more as

we have discussed in Section 3.1.4. So in the authentication procedure, the cloud should not possess any identity information or any secrets about the users. Instead, if the cloud wants to authenticate the users, it must only rely on some publicly available information. Moreover, during the authentication procedure, the cloud is unable to get any useful information to discover or deduce the privacy of the users.

5.2 Setup of Secret Key

5.2.1 Preparation of Setup

Each user of the pseudonym system needs to set a main secret key which is denoted as “Major Secret Key” (MSK). MSK is only known by the user. The secret key is very important to the user, because it is the unique identifier for the user and all the private information of the user will be related with MSK. So the user must carefully choose and keep it.

To set up MSK in practice, a setup software will instruct users. So the users do not have to know the details of the algorithms for MSK setup. A user’s MSK is initially set when the user obtains a blank smart card (we assume the use of smart cards in the pseudonym system to identify users). The setup software asks some necessary inputs (e.g. a string of randomly chosen letters) from the user to generate a secure MSK automatically. The generated MSK is stored in the protected memory of the user’s smart card. Meanwhile, a password denoted as PIN (PIN is usually short and easily memorable like numbers) also must be set by the user to avoid abuse of the smart card. In many existing solutions for smart card, to avoid the brute force guessing of the PIN, the PIN can be protected by another super password to recover or reset the PIN (Lassus, 1997). If a user inputs PIN wrongly for several times, the smart card will be locked and the super password is needed to unlock it (Deo et al., 1998).

5.2.2 Algorithm for Setting up Secret Key

A user chooses a k -bits (k is usually no less than 160) prime integer q , and another prime number p ($p > q$) which satisfies $q | (p - 1)$. By Z_p^* we denote a multiplicative group modulo p . The user finds $g \in Z_p^*$, to be of order q modulo p . Then g is the generator of the cyclic subgroup G_q . By randomly choosing $x < q$, the user’s MSK is formed: $[x, g, p, q]$. The details of the algorithm for generating the user’s MSK is presented in the following Algorithm 1.

In Algorithm 1, the user’s input is not distinctly shown. Actually choosing the rand values (e.g. q and g) in the algorithm may require some arbitrary inputs from the user. However, the user does not need to remember these arbitrary inputs because they are useless after the generation of MSK.

In some systems, the parameters of G_q are global, i.e., each user can use the same g , p and q . Thus, the setup of a user’s MSK is just to choose a random x , which is much easier than the process in Algorithm 1. However, each user could also choose personal parameters to enhance the security (e.g. a case introduced in Section 6.1.7.3).

Algorithm 1: Generating the user's MSK

INPUT: NONE
OUTPUT: MSK

WHILE (true)

 Choose a random q with required length (e.g. 160 bits) IF q is not prime THEN

CONTINUE

 $p = 2 * q + 1$ IF p is prime THEN

BREAK

WHILE (true)

 Choose g randomly less than p IF $g^q \bmod p = 1$ THEN

BREAK

 Choose x randomly less than q RETURN MSK = $[x, g, p, q]$

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

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

For ease of notation, we will sometimes drop the “mod p ” part of the arithmetic expressions in G_q . We build up our secure pseudonym scheme based on the discrete logarithm problem in the cyclic subgroup G_q .

5.3 Algorithm for Generating Pseudonyms

We denote PID_0 as (a_0, b_0) which is $(0, 0)$ by default (PID_0 could be some variable initial values to avoid possible pseudonym collisions and security risk, which will be discussed in Section 6.1.7). PID_0 is the initial pseudonym of the user, but it will never be used elsewhere except being used as an initial value for generating other pseudonyms.

Assuming that previously a user has already generated i PIDs (i.e., $PID_0, PID_1, PID_2, \dots, PID_i$), the $(i+1)^{\text{th}}$ pseudonym PID_{i+1} can be generated by the following Algorithm 2.

Algorithm 2: Generating the user's pseudonym

INPUT: $PID_0 = (a_0, b_0)$, i , MSK, PIN (input by the user to enable the smart card)

OUTPUT: PID_{i+1}, EK_{i+1}

$$EK_{i+1} = KHash(i + 1 || a_0 || b_0, x)^*,$$

where $KHash$ is a keyed hash function with key x

$$a_{i+1} = g^{EK_{i+1} + Hash(a_0 || b_0) \bmod q}$$

$$b_{i+1} = a_{i+1}^x$$

$$PID_{i+1} = Hash(i + 1 || a_{i+1} || b_{i+1})$$

* $||$ denotes the bit concatenation.

In Algorithm 2, the $KHash$ and $Hash$ can adopt any current cryptographic hash functions like SHA-2. In the computation of a_{i+1} , the hash values are converted into a big integer, where the bit-string of hash value is considered as an unsigned binary integer.

We use a keyed hash to simply generate an encryption key corresponding each pseudonym. The inputs of $KHash$ i.e. $(i + 1 || a_0 || b_0, x)$ guarantee that the encryption keys from one user or from different users have trivial probability to collide. The following computation of a_{i+1} and b_{i+1} aims at the authenticity of pseudonyms, which will be shown in Section 5.5. The last hash function $PID_{i+1} = Hash(i + 1 || a_{i+1} || b_{i+1})$ aims to output a pseudonym with reasonable size, and to make the probability of collision of two pseudonyms from one user trivial (refer to the security proof in Section 5.6.1).

Algorithm 2 must be executed inside the user's smart card protectively due to the using of MSK. However, to decrease the computation load of the smart card, some computation can be aided by the coupling device (e.g. a card reader or computer). For more details about the computation and performance consideration in Algorithm 2, please refer to Section 7.4.3.

Because the serial number of the last used pseudonym, i.e. the number i , is needed to generate the next new pseudonym PID_{i+1} , the serial number of the last generated pseudonym should be stored somewhere, e.g., on the user's smart card. As an output of Algorithm 2, EK_{i+1} can be used as the encryption key (it might be necessary to be truncated or padded according to the encryption algorithm) to encrypt the private data related with the new pseudonym PID_{i+1} (e.g. the pseudonym PID_{i+1} acts as the index of the private data).

Alternatively, to decrease the computation of modular exponentiations in Algorithm 2 for the consideration of performance in some resource limited applications (refer to chapter 7 for more discussion on performance), we propose an optional algorithm to generate pseu-

donyms as presented in the following Algorithm 3. The differences regarding security of Algorithm 2 and Algorithm 3 will be discussed in Section 6.1.7.6.

Algorithm 3: An optional algorithm for generating a user's pseudonyms

INPUT: $PID_0 = (a_0, b_0)$, i , MSK, PIN (input by the user to enable the smart card)

OUTPUT: PID_{i+1} , EK_{i+1}

IF i is zero *THEN*

$$EK_{i+1} = KHash(i + 1 || a_0 || b_0, x),$$

where $KHash$ is a keyed hash function with key x

$$a_{i+1} = g^{EK_{i+1} + Hash(a_0 || b_0) \bmod q}$$

$$b_{i+1} = a_{i+1}^x$$

$$PID_{i+1} = Hash(i + 1 || a_{i+1} || b_{i+1})$$

ELSE THEN

$$EK_{i+1} = KHash(i + 1 || a_0 || b_0, x)$$

$$PID_{i+1} = Hash(i \wedge a_1 || i \wedge b_1), \text{ where } \wedge \text{ is bit xor operation, and } i \text{ is represented}$$

as bit string with the same length as a_1 and b_1 .

5.4 Algorithm for Reproducing Pseudonyms

All pseudonyms and encryption keys of a user can be reproduced one by one through the following Algorithm 4. Similar to Algorithm 2, Algorithm 4 must also be executed inside the user's smart card due to the using of MSK.

Algorithm 4 illustrate the procedure to reproduce a user's pseudonym and the encryption key with any serial number i . In fact, any single PID_i and EK_i pair can be reproduced independently with other pseudonyms as shown in the above Algorithm 4. For example, to reproduce PID_{100} , we do not need to precompute PID_1 to PID_{99} . Instead, we just set $i=100$ to run Algorithm 4 and get the computed pseudonym PID_{100} .

Algorithm 4: Reproducing the user's pseudonym

INPUT: $PID_0 = (a_0, b_0)$, MSK, PIN, i , $last$ (the serial number of last used pseudonym PID_{last})

OUTPUT: PID_i, EK_i

IF $i > last$ *OR* $i < 1$ *THEN* return null.

ELSE

$$EK_i = KHash(i || a_0 || b_0, x),$$

$$a_i = g^{EK_i + Hash(a_0 || b_0) \bmod q}$$

$$b_i = a_i^x$$

$$PID_i = Hash(i || a_i || b_i)$$

Corresponding to the Algorithm 3, reproducing a user's pseudonyms and encryption keys optionally is presented in Algorithm 5.

Algorithm 5: An Optional way of reproducing a user's pseudonyms

INPUT: $PID_0 = (a_0, b_0)$, MSK, PIN, i , $last$ (the serial number of last used pseudonym PID_{last})

OUTPUT: PID_i, EK_i

IF $i > last$ *OR* $i < 1$ *THEN* return null.

IF $i == 1$ *THEN*

$$EK_i = KHash(i || a_0 || b_0, x),$$

$$a_i = g^{EK_i + Hash(a_0 || b_0) \bmod q}$$

$$b_i = a_i^x$$

$$PID_i = Hash(i || a_i || b_i)$$

ELSE THEN

$$EK_i = KHash(i || a_0 || b_0, x),$$

$$PID_i = Hash(i \wedge a_1 || i \wedge b_1) \text{ where } \wedge \text{ is bit xor operation}$$

5.5 Authentication of Pseudonyms

Following is the algorithm for a verifier (or authenticator), e.g. the cloud (abbreviated as C), to validate a user's (abbreviated as P) ownership of some private data related with a pseudonym PID .

Algorithm 6: Algorithm for authentication of PID

INPUT: a PID in the cloud; p, q in the user's MSK are known by the cloud

OUTPUT: Yes or No.

P → C: User sends (i, a, b) such that $PID = Hash(i||a||b)$ *

C: checks whether $PID \stackrel{?}{=} Hash(i||a||b)$. If not, C returns No; otherwise continues.

P → C: User randomly chooses s , calculates and sends $(A=a, B=a^s \bmod p)$

C → P: Cloud randomly chooses and sends c

P → C: User computes and sends $y = s+cx \bmod q$

C: checks $a^y \stackrel{?}{=} Bb^c \bmod p$. If yes, C returns Yes; otherwise returns No.

* "→" means sending a message.

Algorithm 7: An optional algorithm for authentication of PID

INPUT: an PID in the cloud; p, q of the user's MSK are known by the cloud

OUTPUT: Yes or No.

IF PID is the user's first pseudonym *THEN*

Follow *Algorithm 6* for authentication

ELSE

P → C: the first pseudonym PID_1

Follow *Algorithm 6* for authentication on PID_1

P → C: the index number i of PID

C: checks $PID \stackrel{?}{=} Hash(i \wedge a_1 || i \wedge b_1)$, if yes, return true; otherwise false.

Corresponding to the optional pseudonym generation in Algorithm 3, Algorithm 7 is the authentication between the verifier (C) and the user who claim the owner of pseudonym (P). The algorithm for authentication of PID does not provide any protection of the followed communication between the verifier and the user. If the user wants to make some changes to the data indexed by the PID, the changes sent to the verifier (e.g. the cloud) should be protected by other schemes (e.g. encrypted by the public key of the verifier).

In Algorithm 6, there are many message exchanges between the user and the verifier. In some network, the cost of sending and receiving message may be high. We alter Algorithm 6 a little bit as shown in Algorithm 8 to decrease the amount of message exchanges in the authentication procedure, at the price of the verifier sending a big integer c to the user in the beginning of verification.

Algorithm 8: Algorithm for authenticating with less messages

INPUT: PID ; p, q are known by the cloud

OUTPUT: Yes or No.

C \rightarrow P: Cloud randomly chooses and sends c

P \rightarrow C: User computes (a, b) such that $PID = Hash(a||b)$;

User randomly chooses s , calculates $(A=a, B=a^s \bmod p)$;

User computes $y = s+cx \bmod q$;

User sends (a, b, A, B, y)

C: Checks whether $PID \stackrel{?}{=} Hash(a||b)$.

If not, C returns *No*; otherwise continues.

checks $a^y \stackrel{?}{=} Bb^c \bmod p$. If yes, C returns *Yes*; otherwise returns *No*.

5.6 Security Evaluation

The security of the above algorithms relies mainly on the difficulty of the discrete logarithm problem in G_q and the one-way property of the hash function. We prove briefly the security of these algorithms in this Section. In the following proof, we assume that the outputs of hash and keyed Hash have the length of 256 bits, and we also assume that the parameters for secret keys are generated according to Algorithm 1, which generates two secure primes (p, q) and a difficult discrete logarithm in a specified subgroup. Thus we assume that solving such a discrete logarithm problem has a complexity of $O(2^{256})$ if the size of the q is 512 bits (Adrian et al., 2015, Shoup, 1997, Lim and Lee, 1997).

5.6.1 Proof of Algorithm 2

One way of pseudonyms. In Algorithm 2, if an attacker wants to reverse the pseudonym to obtain the secret key (the number x) of the user, he firstly needs to find out the first preimage (i.e. a_{i+1} and b_{i+1}) of PID_i in the hash function (Rogaway and Shrimpton, 2004). The complexity of first preimage attack on a delicate hash function is $O(2^{256})$. Then the attacker need to solve the DLP to find x in $b_{i+1} = a_{i+1}^x$. The complexity of solving this is also $O(2^{256})$. So in total the complexity to reverse the pseudonym of the user is $O(2^{256})$.

Collision-free of pseudonyms. If we assuming that two different users have different secret x or different (a_0, b_0) as discussed in Section 6.1.7.4, two pseudonyms (from a same user or two users) must have different parameters for calculating EK_{i+1} . I.e., one user calculates different EK_{i+1} with different i , two different users use different x or (a_0, b_0) . According to the collision features of hash function, two arbitrary EK_{i+1} collide with probability of 2^{-256} . If a user has the same EK_{i+1} with different serial number i , as a consequence the same a_{i+1} and b_{i+1} will be computed. However, because the last step $PID_{i+1} = Hash(i + 1 || a_{i+1} || b_{i+1})$ in Algorithm 2 have different inputs (i.e. $i+1$), the two pseudonyms collide with the probability of 2^{-256} . If two different users calculate the same EK_{i+1} , they must have different a_{i+1} and b_{i+1} computed because they have different x or (a_0, b_0) . As a result the two users get the same $PID_{i+1} = Hash(i + 1 || a_{i+1} || b_{i+1})$ with the collision probability of 2^{-256} according to the feature of hash function. In one word, the collision of two arbitrary pseudonyms have the probability of 2^{-256} . Or we say that a collision happens in 2^{128} pseudonyms with probability of $\frac{1}{2}$. Some systems have very strict requirement to the collision of pseudonym. We discuss a case to solve the occasional collision in Section 6.1.7.4.

Independence of pseudonyms. Due to the hash function used in $PID_{i+1} = Hash(i + 1 || a_{i+1} || b_{i+1})$, the pseudonyms of a user are pseudo random and independent without knowing the secret key x . The complexity of finding the user's other pseudonym from knowing one or more pseudonyms has the same complexity as finding the user's secret x , i.e., $O(2^{256})$.

5.6.2 Proof of Algorithm 3

The one-way and collision-free features of Algorithm 3 can be similarly proved as Algorithm 2. However, the independence of pseudonyms is not as strong as Algorithm 2. In Algorithm 3, a user's pseudonyms are derived from PID_1 by a hash function. If PID_1 are not know by attackers, all the pseudonyms are still independent. However, if the PID_1 is disclosed, all other pseudonyms of the user are can be computed by attackers.

5.6.3 Proof of Algorithm 6, 7, 8

The Algorithm 7 and 8 are identical to Algorithm 6. In the following we only prove the security of Algorithm 6.

The cloud or the attacker who can eavesdrop the authentication process can obtain (i, a, b, c, y, A, B) . An attacker wants to get the secret key x of a user according to the obtained

information. Obviously, from knowing a and b , the complexity of computing x has the complexity of $O(2^{256})$ according to the one-way proof in Section 5.6.1. In $y = s+cx \pmod q$, there are two unknown elements s and x , when y and c are known. To compute x , s has to be computed from $B=A^s \pmod p$. The computation has complexity of $O(2^{256})$. In one word, the attacker has the complexity of $O(2^{256})$ to compute the secret key of x from knowing any authentication messages.

An attacker who wants to impersonate the user has to answer the verifier the correct y and A, B which satisfy $A^y = Bb^c$. If the verifier generates the challenge number c randomly (to avoid simply replay attack), the attacker who does not know x has to solve the discrete problem $A^y = D$ where A and D are known in order to get correct y , which has complexity of $O(2^{256})$.

6 Applications of the Pseudonym Solution in eHealth Systems

6.1 Application in Ordinary Healthcare Activities

6.1.1 Cloud-based eHealth System with Pseudonyms

We have proposed a model of cloud-based eHealth systems in Figure 9. With the introducing of pseudonym solution, a cloud-based eHealth system with major ordinary healthcare activities is shown in Figure 10.

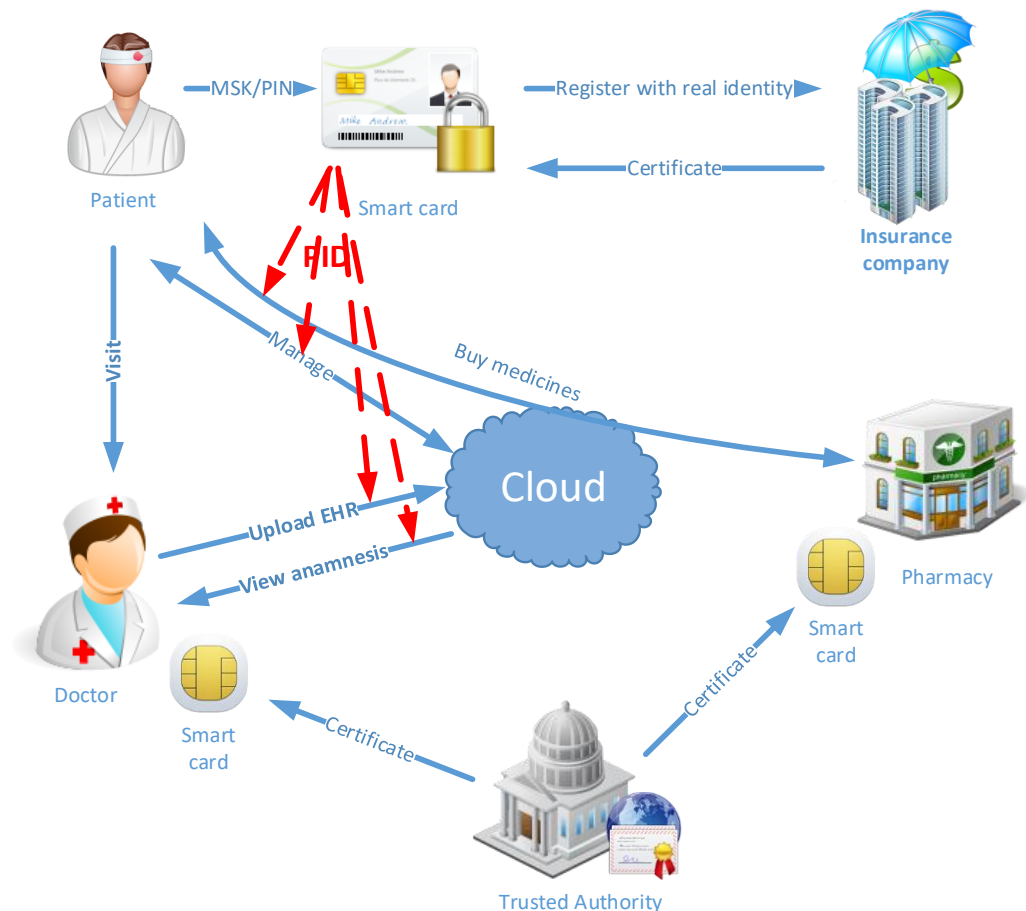


Figure 10: Using pseudonyms in the ordinary healthcare activities

The cloud plays the role of associating almost all participating entities such as patients, doctors, pharmacists, and health insurance companies (will be abbreviated as insurance

companies) in an eHealth system. The pseudonym solution can benefit these ordinary healthcare activities with featured security and privacy protection. The following paragraphs will generally introduce the settings to adapt the proposed pseudonym solution in chapter 5 in to a cloud-based eHealth system. Section 6.1.2 to 6.1.6 will present the details of how to protect the security of health data and the privacy of patients in some ordinary healthcare activities.

As shown in Figure 10, each patient has a set of secrets consisting of MSK and PIN, which are only known by the patient himself. MSK is the main secret key of the patient and stored in the protected memory of a smart card, for which a PIN is employed to authenticate the patient to avoid abuse as introduced in Section 5.2. MSK of the patient is used to generate pseudonyms, encrypt EHR, prove the patient's ownership on EHR entries to the verifiers, and for even more functions (e.g., the applications introduced in Section 6.2 and 6.3). Because MSK is tightly related to the security of EHR and the privacy of the patient, the patient has the responsibility and incentive to keep them safe. I.e., MSK should never be accessed or used by someone else other than the patient himself. In our solution, a smart card is employed to protect MSK by secure hardware (Grand, 2004). Moreover, in case of losing the smart card, MSK should be safely backed up by the patient at somewhere other than in the smart card, e.g. in the patient's computer. We will discuss the issues and countermeasures of smart card loss in Section 6.2.3.3.

The pseudonyms (PID) of a patient will be used in most ordinary healthcare activities as shown in Figure 10. Contrarily, each patient's real identity is only known by the insurance company while the patient registers there in the beginning as introduced in section 3.3.2.1. In theory, the real identity of the patient is not necessarily known by the insurance company. However, in many countries, the insurance premium is paid partly by the employer and partly by the patient. Moreover, the patient has to contact an insurance company in person to sign the insurance contract. So, the identity of the patient is difficult to be a secret against the insurance company in practice. The special aspects of protecting patients' privacy against insurance companies are discussed in section 6.2. In fact, the application of the pseudonym solution proposed in this thesis does not depend on the existence of insurance companies. It is easy to adapt the pseudonym solution in eHealth systems without an insurance company, which actually brings troubles to the deployment of the pseudonym solution as to be discussed in section 6.2.

After registering at a health insurance company, each patient will get a digital certificate from the insurance company with a corresponding private key which is only known by the patient (we will denote the patient's corresponding private key from the insurance company as SKI). The patient's identity (e.g. name, address) is not included in this certificate. Typically, a certificate number, which can only be mapped to the patient's real identity by the insurance company, a token (e.g. company name, registration number) of the insurance company, the public key of the patient, and the validity period are necessary in the patient's certificate.

As we have introduced in Section 3.3.1.1, the healthcare providers (e.g. doctors and pharmacists) need to get certificates from a trusted authority with self-known private keys. The trusted authority could be the health department of the government which confirms healthcare providers' qualifications by issuing them digital certificates, in which information like healthcare categories and healthcare providers' identities can be enclosed. The certifi-

cates and the corresponding private keys can be stored in the protected memory of smart cards issued to healthcare providers with authenticating PINs. Healthcare providers' private keys will be used to sign the record sections of the patient's EHR, prescriptions and bills, which will be separately introduced in Section 6.1.2 and 6.2. As a comparison, the private keys of patients are only used for signing the bills. The billing procedure is not depicted in the Figure 10 for simplicity. For more details of the bills relating to patients' SKI, please refer to Section 6.2.

We will describe how to utilize the pseudonym solution in some major ordinary healthcare activities for the protection of security and privacy in the following Sections from 6.1.2 to 6.1.6.

6.1.2 Concealing Patients' Identities and Protecting the Security of EHR

A typical pseudonym generation (the execution of Algorithm 2) occurs at a doctor's practice when a patient visits the doctor, taking along his smart card as shown in Figure 10. The doctor checks the validity of the patient firstly by asking the patient to input the PIN of the smart card. If necessary, the doctor can further check the patient's certificate and challenge the patient's SKI, with the help of the insurance company's root certificate which could also be issued by the trusted authority. If all validations are successful, the doctor (or any other staff in the practice) diagnoses, examines, treats the patient, and then writes down a record (including the diagnosis, examinations, treatment and other information) and probably a prescription stating the medicines that the patient needs to buy at a pharmacy. Meanwhile, a new pseudonym (PID) and an encryption key (EK) are generated by the patient's smart card from MSK by using the Algorithm 2 or Algorithm 3. Both the record and the prescription will be encrypted by EK and signed by the doctor's private key. Then, all the data from the doctor are uploaded to the cloud along with an index header, the new PID, forming a new EHR entry for the patient's EHR. Thus, an EHR entry needs to be at least compatible with the following segments where "||" means concatenation,

EHR entry =

$$\text{PID} \parallel \text{Enc}_{\text{EK}}(\text{record}) \parallel \text{sig}_d(\text{record}) \parallel \text{Enc}_{\text{EK}}(\text{prescription}) \parallel \text{sig}_d(\text{prescription}).$$

In the EHR entry, $\text{Enc}_{\text{EK}}(\dots)$ is the cipher text of the record or the prescription by using the encryption key EK. Any secure cryptographic cipher scheme (e.g. AES) can be utilized for the encrypting. Importantly, the encrypting operation is executed by the patient's smart card, because EK should not be disclosed to anybody else except the patient. Even the doctor does not know the patient's encryption key. EK can also be used to encrypt some extra data (e.g. the access control table on EHR entry, and the communication with secondary users introduced in Section 6.3.) which are not necessarily known by the doctor. $\text{sig}_d(\text{record})$ and $\text{sig}_d(\text{prescription})$ are the signatures of the doctor on the record and prescription.

Due to the use of the pseudonyms and encryption, the EHR of a patient do not contain any real identifiers (e.g. name, identity number, age, gender.) in plain text. These identifiers are replaced by the pseudonyms which securely conceal the identity of the patient. Moreover, as we have introduced in the last paragraph, the doctor does not necessarily know the real identity of the patient. But in practice, since the doctor is assumed to be trusted, it is usual-

ly the case that the identifiers (at least parts of identity information such as gender, age, and telephone number) of the patient is inevitably disclosed to the doctor.

6.1.3 Indexing EHR (Electronic Health Records) Entries

The cloud stores EHR entries of all patients by indexing the PID segments of all EHR entries. In order to avoid illegal uploading of EHR entries (e.g. attackers may upload useless data to form DoS attack), the cloud validates uploaders (doctors) by checking their certificates and challenging their private keys with the help of the trusted certificate authority. From the viewpoint of the privacy of patients, this validation is actually unnecessary, because any illegally uploaded (forged) EHR entries can be easily recognized by the patients due to the following reasons. Without the authorization (by inputting the PIN) of a patient, nobody else can generate a valid encryption key or PID of the patient because MSK is needed for generating PID and EK according to Algorithm 2. Moreover, each pair of valid PID and EK is only used for one EHR entry as shown in Algorithm 2 or Algorithm 3. Thus, the forged EHR entries (e.g. an EHR entry with a known PID of the patient) can easily be recognized and removed by the patient by checking the correctness of decryption and the PID.

The cloud does not keep any registration information of patients. Hence, it does not know the identities of patients except the PIDs (the PIDs will be used to authenticate patients as introduced in Section 6.1.5). Moreover, because the sensitive contents in EHR entries are encrypted with the encryption keys which are unknown to the cloud, the cloud is unable to disclose the identities of patients or any meaningful information in EHR, even if the cloud is curious to know. For the same reason, any attackers who can access all EHR entries in the cloud can not get any useful information from EHR or discover the identities of patients.

6.1.4 Retrieving EHR Entries

The PIDs are reproduced by patients when EHR entries are needed to be retrieved. For example, when a doctor who is examining a patient wants to view the anamnesis of the patient or when a patient wants to view and manage his own EHR at home, the patient regenerates the PIDs one by one using Algorithm 4 or Algorithm 5 and sends them to the cloud in order to download the content of the corresponding EHR entries. The cloud simply returns all EHR entries indexed by the requested PIDs.

It is easy to retrieve all EHR entries of a patient but sometimes it is not the desired way. Supporting query conditions (e.g. specified date and illness type) is helpful for the inquirers (e.g. a doctor) to get a particular subset of EHR entries rather than all EHR entries of the patient. Even sometimes, the inquirers just need partial segments (e.g., the general information of the illnesses of the patient) of EHR entries, because the size of the whole EHR entries is too big to download from the cloud in short time. In that case, the cloud needs to have the ability to search information in the EHR. Moreover, the structure information of EHR entries has to be disclosed to the cloud. Searching in the encrypted EHR is necessary to support such complex inquiring, which is another research topic in cryptology and eHealth systems (refer chapter 8 for more discussion). In the setting of the cloud-based eHealth system in this thesis, because the cloud does not know which pseudonyms belong to one patient and has no knowledge about the encryption keys of EHR entries, patients

themselves have to be involved in the construction of encrypted searching index. They must cooperate with the cloud to enable complex searching in the encrypted EHR. The encryption algorithm in our solution is still quite open to any searchable encryption schemes, although we implemented the encryption by block cipher (AES) in our prototype introduced in chapter 7. Our implementation currently does not support the search in encrypted EHR entries, but we suggest to use partial encryption to support fast search in unencrypted data (refer Section 6.3 for more details).

6.1.5 Authenticating the Ownership of EHR

Pseudonyms are used to authenticate patients' ownerships on their EHR entries. When a patient wants to manage/change his own EHR entries (e.g. to make some notes or add access control properties), the cloud has to verify that the patient is the owner of those EHR entries beforehand to avoid malicious modifications. In our system model, in order to avoid the risk of disclosing patients' identities, EHR entries do not contain patients' certificates or signatures as introduced in Section 6.1.2, and the cloud does not keep any registration information about the patient. To our delight, the proposed pseudonym solution provides a way for the cloud to authenticate patients' EHR ownerships without knowing or leaking the information of patients' identities as presented in Algorithm 6 or Algorithm 7. Because each EHR entry is indexed (related) by a PID, the cloud can check the ownership of the patient on the EHR entry by authenticating the patient's ownership of the PID. After the cloud authenticating the patient, the patient can send the changes to the cloud. It is advisable that the changes are sent in a secure channel between the patient and the cloud, to avoid eavesdrops and other attacks. For example, the changes could be encrypted by the public key of the cloud, thus the encrypted changes can only be decrypted by the cloud.

6.1.6 Purchasing Medicines from Pharmacy

Another typical scenario for updating an EHR entry is when a patient goes to a pharmacy and buy medicines by using the prescription which is enclosed in the EHR entry as described in Section 6.1.2, the EHR entry should be marked that the prescription is used (or how many times are used out of the allowed times) in order to prevent the abuse of the prescription.

The patient retrieves the EHR entry from the cloud by providing the cloud the corresponding PID (the PIDs of EHR entries containing unused prescriptions can also be temporarily stored in the patient's smart card for fast query). The EHR entry is then decrypted by corresponding EK. The decrypted prescription with the doctor's signature is shown to the pharmacist. The pharmacist checks the doctor's signature on the prescription. If the signature is valid, the pharmacist generates an additional signature on the prescription to indicate that the prescription has been used (or once). Then the patient must update the original prescription's signature segment of the EHR entry by adding the pharmacist's signature (note that the signature segment is not encrypted!). To avoid illegal updating, the cloud needs to check beforehand whether the updater (the patient) is the owner of EHR entry or not. Only if he is, the cloud updates the old prescription's signature segment by adding the pharmacist's signature. After the pharmacist confirms that the patient has updated the prescription signature (by reading the signature segments in the cloud), the medicines are then sold to the patient.

In this procedure, we assume that the cloud has knowledge about the structure of EHR entry, and only allows the patient adding new signatures to the prescription's signature segment (i.e., the patient is not allowed to delete a signature). In order to avoid reusing the prescription, before accepting the prescription the pharmacist must check whether there is already another pharmacist's signature existing in the prescription's signature segment or not. In practice, a prescription may be used for several times, e.g. in case of chronic diseases. We can equip the prescription with a property indicating how many times it can be reused. This property is determined by the doctor who wrote the prescription. The pharmacist can read this property in the decrypted prescription and check how many times the prescription has so far been used (i.e. how many other pharmacists' signatures exist) to determine whether the prescription is still valid or not to be used once again. We assume that the certificates of doctors and pharmacists include the information about their healthcare roles. Thus the pharmacists can distinguish the signers of the signatures (from a doctor or a pharmacist).

Due to the above updating in the procedure of medicine purchasing with prescription, the updated EHR entry with a pharmacist's signature looks like as follows,

$$\text{updated EHR entry} = \text{PID} \parallel \text{Enc}_{\text{CEK}}(\text{record}) \parallel \text{sig}_d(\text{record}) \parallel \text{Enc}_{\text{CEK}}(\text{prescription}) \\ \parallel \text{sig}_d(\text{prescription}) \parallel \text{sig}_p(\text{prescription}),$$

where the segment $\text{sig}_p(\text{prescription})$ denotes the signature of a pharmacist on the prescription. For the meaning of other segments, please refer to Section 6.1.2.

6.1.7 Potential Threats and Countermeasures

6.1.7.1 Cloud's Knowledge of the PIDs of Patients

Because all pseudonyms of a patient are independent without knowing the patient's secret keys, common attackers are unable to know which EHR entries (indexed by different PIDs) belong to one patient. However, the cloud may know more about the pseudonyms of one patient than the common attackers outside. A patient's frequent access to his EHR may potentially disclose all PIDs of the patient to the cloud, and a doctor treating the patient may have to access all EHR entries of the patient from the cloud. Under these cases, all pseudonyms of the patient may be sent to the cloud for retrieving all the corresponding EHR entries in one query. If the cloud wants to, it can record the pseudonym sets from different querying sources (e.g., network IP). The pseudonyms in one set come from one patient with high probability.

This knowledge of the cloud does not harm the privacy of patients directly, because the real identities of patients are unknown to the cloud and the essential segments of EHR are encrypted. However, if this pseudonym set knowledge is disclosed to a close attacker who is familiar with a patient and knows the patient's identifiers and even some extra information of the patient's healthcare activities, the attacker may discover some general information about the patient's illness. For example, if the close attacker happens to know that a familiar patient has visited doctor A and doctor B recently. With the knowledge of all patients' pseudonym sets, the attacker can find out the patients who have visited both A and B from the doctor's signature segments of EHR entries. This will narrow the range of

candidates and even help to find out the exact pseudonym set that the familiar patient has. The attacker can download all EHR entries indexed by the PIDs in the pseudonym set from the cloud. Then the close attacker may discover some general information about the patient's illness type from the doctor's signature segments of these EHR entries, because the doctor's category can be easily obtained from the doctor's certificate which is enclosed in the signature.

A direct countermeasure is to move the doctor's certificate into the record segment of EHR entry. This can prevent the close attacker from knowing the doctor's category in certificate, i.e., the illness type of the patient, because the record segment with the doctor's certificate is encrypted. However, as introduced in Section 6.1.3, the cloud is able to obtain the certificates of doctors when they are authenticated by the cloud to upload new EHR entries, even if doctors' certificates are enclosed into the encrypted record segment. Fortunately, the cloud does not have the close attacker's extra information about the patient. Thus, the pseudonym set knowledge does not interest the cloud.

In the worst case, the cloud may collude with the close attackers to disclose the patient's privacy, i.e., the cloud discloses the pseudonym set information to the close attackers. Hence, it is still valuable to prevent the cloud from obtaining the pseudonym set knowledge. Some tricks can be used to blur the knowledge of exact pseudonym sets. E.g., when a patient or doctor retrieves EHR entries from the cloud, some fake pseudonyms can be sent besides the intended pseudonyms. The fake pseudonyms mean those pseudonyms existing in the cloud but belonging to other patients. We assume that the pseudonyms in the cloud are public available resources, and anyone can view these pseudonyms freely. Another solution is that patients and doctors can use anonymous proxies (Reed et al., 1996) and particular network technologies to avoid being traced by the cloud.

6.1.7.2 Protecting Identities against Eavesdroppers

When a patient accesses his/her EHR from the computer at home or from a mobile device, there exists risk of mapping the pseudonyms to patients' identities resulting from ISP (Internet Service Provider) and eavesdroppers. In many countries, the customers' real identities are known by the ISP. Moreover, the ISP knows which network address (IP address) patients are using. As a result, if the ISP wants to, it can easily record the pseudonyms that a patient sends to the cloud. Some attackers who can eavesdrop on the patient's network communication can also record the pseudonyms transmitted. An easy way to prevent the ISP or the eavesdroppers from knowing the pseudonyms of patients is to use a secure channel (e.g. using SSL) (Viega et al., 2002) for transmitting unencrypted data (e.g. PIDs) between patients and the cloud.

6.1.7.3 Avoiding Guessing Attacks against Secret Keys

A patient's main secret key MSK may suffer from brute guessing attack which tries to discover the patient's secret keys by random guessing. As introduced in the chapter 5, a patient's choice of x in his MSK is usually a large integer with a length of more than 160 bits. An attacker tries a random x' and computes the first pseudonym by Algorithm 2 or Algorithm 3. Then the pseudonym is sent to the cloud to ask whether a corresponding EHR entry indexed with this pseudonym exists or not. If exists, the attacker has successfully guessed the MSK of one patient in the eHealth system. The attacker only has a probability

of $1/2^{160}$ to guess the right x in MSK by one random attempt if the size of x is 160 bits. However, under the setting of the cloud-based eHealth system in our thesis, the attacker may have better luck than random guess. Imagining that there are N patients in the system, the probability of a success guess will be $N/2^{160}$ because there are N such first pseudonyms in the cloud. It is not practical for the attacker to succeed in the attack when N is not big. However, the probability is much higher than the theoretical value $1/2^{160}$ if N is large. A successful guess of one patient's secret key may cause serious consequences because the attacker can view all EHR entries of the patient and even impersonate the patient.

There are some ways to rule the attack out when N is considerably large. One solution is to enable each patient to freely choose g , p , and q besides x in MSK. This will increase the computation a little when a patient initializes his smart card for setting up MSK as presented in Algorithm 1, but it can add a big burden to the attacker. The attacker has to guess g , p , and q which are also big numbers besides x . This will decrease the success probability of the attacker's guess to a negligible value. Another way is to make use of PID_0 . In the algorithm of the pseudonym generation in Algorithm 2 or Algorithm 3, PID_0 is an initial parameter and is set to $(0, 0)$ by default. To prevent attackers from guessing patients' secret keys, each patient can use a different PID_0 . Thus, the attacker needs to guess PID_0 besides x . It will also make the success probability of the attacker's guess drop to a negligible value.

6.1.7.4 Dealing with Pseudonym Collision

In the pseudonym solution presented in chapter 5, the secret parameters (e.g., x) in MSKs for generating pseudonyms of patients are chosen by patients themselves separately. It is probable that two innocent patients happen to choose a same parameter set, which results in that the two patients have the same pseudonyms. Although the probability is very low due to the large space of the parameters, the collision of pseudonyms would certainly cause serious problems if happened. Because the pseudonyms generated by the two patients will be equal, and it will confuse EHR entries from the two patients. A solution is to utilize PID_0 to avoid the occurrence. The manufacturer of the smart cards can control that all patients have different PID_0 's, or the PID_0 can be set by each patient to be (or the hash value of) the patient's unique identifiers (e.g., identity no., passport no., or SSN). According to the pseudonym generation in Algorithm 2 or Algorithm 3, it is expected that different pseudonym sequences will be produced due to different PID_0 .

There is another case of collision where two arbitrary pseudonyms (from one patient or two patients) might collide because Algorithm 2 or Algorithm 3 is not injective, i.e., two equal pseudonyms can be produced although different inputs are used in the algorithm of pseudonym generation. The probability of the collision is determined by the birthday paradox (Flajolet et al., 1992). For example, if the bit length of each pseudonym is n bits, a collision of two pseudonyms happens with the probability of $\frac{1}{2}$ if there are $2^{n/2}$ pseudonyms in the eHealth system. The collision of pseudonyms may also lead to the confusion on the EHR entries indexed by collided PIDs. A direct solution is that the cloud forbids the collision: when a patient uploads an EHR entry with collided PID, the cloud will ask the patient to provide a new one. The patient will remember in his smart card that this PID can not be used, which requires the smart cards to provide extra storage for collided PIDs. Another better responding solution is to count on the encryption or keyed HMAC (Keyed Hashing for Message Authentication Code) on EHR entries. That is, when the pseudonyms in two

EHR entries happens to collide, the two EHR entries must have been encrypted or HMACed with different secret keys (e.g., EK). So, in the procedure of decrypting a collided EHR entry or the verification of HMAC, the EHR entry that can be successfully decrypted or has the correct HMAC is the correct one. In our design of Algorithm 2 and Algorithm 3, two EHR entries with the same pseudonym are highly expected to have different encryption keys or different HMACs because the formulas for outputting PID and EK have different parameters. Thus, a patient can recognize which EHR entry is the correct one if he receives more than one answer from the cloud after he sent one pseudonym to the cloud. The collided EHR entry which cannot be correctly decrypted or cannot get a correct HMAC is considered as undesired EHR entries by patients.

Due to the collisions of pseudonyms, the authentication in Algorithm 6 or Algorithm 7 might be also misused. For example, in the updating procedure of an EHR entry with a collided PID (two EHR entries E1 and E2 separately from two different patients P1 and P2 are indexed by a some PID) as introduced in Section 6.1.5, because the cloud cannot tell which patient is the exact owner of the two EHR entries, P1 could update P2's EHR entry E2 (P1 can prove to the cloud that he is the owner of the pseudonym of E2 in Algorithm 6 or Algorithm 7). One simple solution is that a patient must create a new EHR entry with reference to the updated entry, when he wants to update a EHR entry. This can avoid wrong updating because the original EHR entries keeps unchanged. Another solution relies on the cloud to do further checks after the authentication by Algorithm 6 or Algorithm 7. In the case where a patient wants to update the prescription's signature segment of an EHR entry with collided PID as introduced in Section 6.1.6, the cloud requires the patient to decrypt and present the prescription segment, and checks the signature of the doctor or pharmacist. Only if the patient can successfully decrypt the prescription section, the cloud can check the doctor's or pharmacist's signature to furtherly confirm the patient's ownership on the EHR entry. Thus, by the further signature check, the cloud can prevent updating to the wrong EHR entry with collided PID. In Section 6.1.7.1, since we recommend to enclose the doctor's certificate in the encrypted record, the cloud can also ask the patient to decrypt and present the doctor's certificate. Then the cloud can check the validation of the doctor's signature to further verify whether the patient is the real owner of EHR entry or not. In other cases, where the updates are made by the patient and appended to an EHR entry (e.g., to set custom access control), the updates must be encrypted or have integrity protection (e.g. keyed HMAC). The wrong updates can be detected and abandoned by the real owner of the EHR entry though decryption check or integrity check, even if the cloud did not carry out the further checks or wrongly accepted some malicious updates.

6.1.7.5 Trust Mode of Cloud

The cloud in this thesis is not necessary fully trustworthy, but it is expected to be honest as introduced in Section 3.1.2. This is usually the case in practice. Because of the security and privacy threats to the cloud as introduced in Section 3.1.4, the data stored in the cloud are vulnerable for abuse by the cloud itself or easily be obtained by a skilled attacker. Thus, we do not store in the cloud more information (such as patient's registration information) other than protected data and publicly available data (e.g., root certificate of trusted authority for validating doctors).

However, we require that the cloud acts honestly, i.e., the eHealth system deployed by the "system provider" on the cloud is executed honestly. For example, the cloud does not in-

tend to tamper EHR, and it also follows the protocols designed for authenticating the ownership of EHR entries by Algorithm 6 or Algorithm 7. As discussed in Section 6.1.6, we also require the cloud to understand some structure information of EHR entries and to prevent updating to the undesired segments (e.g. the segments of record and prescription) of EHR entries in an honest manner. Due to the feature of the authentication protocols in Algorithm 6 or Algorithm 7, the cloud does not benefit (e.g. to obtain information of patients' identities) from any dishonesty, except disturbing the normal procedure of the healthcare activities.

6.1.7.6 Extra Threats from Optional Algorithms

In chapter 5, we presented Algorithm 3, Algorithm 5 and Algorithm 7 for optional pseudonym solution. These algorithms decrease the computation load on the smart card greatly especially in pseudonym generation and reproducing. E.g., in pseudonym generation of Algorithm 3, only the first pseudonym needs computation of two modular exponentiations which need more computation time, and the other pseudonyms only need several computations of hash functions which are much faster.

These optional algorithms preserve the same security level as Algorithm 2, Algorithm 4 and Algorithm 6, because the security still relies on the DLP and hash functions as introduced in Section 5.6. However, there exists one more risk where they are applied in the ordinary healthcare activities. That is, the cloud is able to obtain the information about each patient's pseudonyms more easily. During the authentication process in Algorithm 7, a patient needs to tell the cloud of his first pseudonym PID_1 and (a_1, b_1) . As a result, the cloud is able to infer all pseudonyms of the patient from Algorithm 5, i.e., the cloud has an easier way to obtaining the pseudonym sets of all patients. The potential risk of the cloud's knowledge on the pseudonym sets of patients was already discussed in Section 6.1.7.1. Nevertheless, the cloud or any other attackers who have such pseudonym set knowledge are not able to generate a forged pseudonym and create the corresponding EHR entry for a patient, because they can not generate the valid EK for encrypting EHR entry. Even if an attacker creates and uploads a fake EHR entry for the patient, the fake EHR entry can be easily detected and excluded by the patient through checking the decryption or integrity by correct EK, because the fake EHR entry must be encrypted or keyed HMACed by an invalid EK.

6.2 Application in Health Insurance

6.2.1 A Refined Billing Procedure

6.2.1.1 Setting in Billing

The billing process is complex and differs in different implementations of eHealth systems. As we have introduced in Section 3.1.3 and 3.3.2.4, in some eHealth systems (e.g. German statutory health insurance systems), healthcare providers (e.g., practitioners, hospitals, pharmacies) send the bills to insurance companies directly to get paid as shown in Figure 4 (a). Patients are not involved in the billing procedure directly. In contrast, some private insurance companies reimburse the bills sent from patients who pay firstly the bills

from healthcare providers as shown in Figure 4 (b). In practice, there are even separate billing companies (consolidators) in some eHealth systems (Milroy and Li, 2001). The bill collectors who work with healthcare providers and insurance companies deal with billing affairs on behalf of them. To simplify the interpretation of our scheme, we present a simple billing model shown in Figure 11. This billing model is a simplified version of many nationwide statutory eHealth systems. We remove some intermediate roles (e.g. bill collectors, banks) in order to describe clearly the essential threats that the privacy of patients confront in the billing procedure. Some issues and workarounds about adopting our scheme in other billing models are discussed in Section 6.2.3.2.

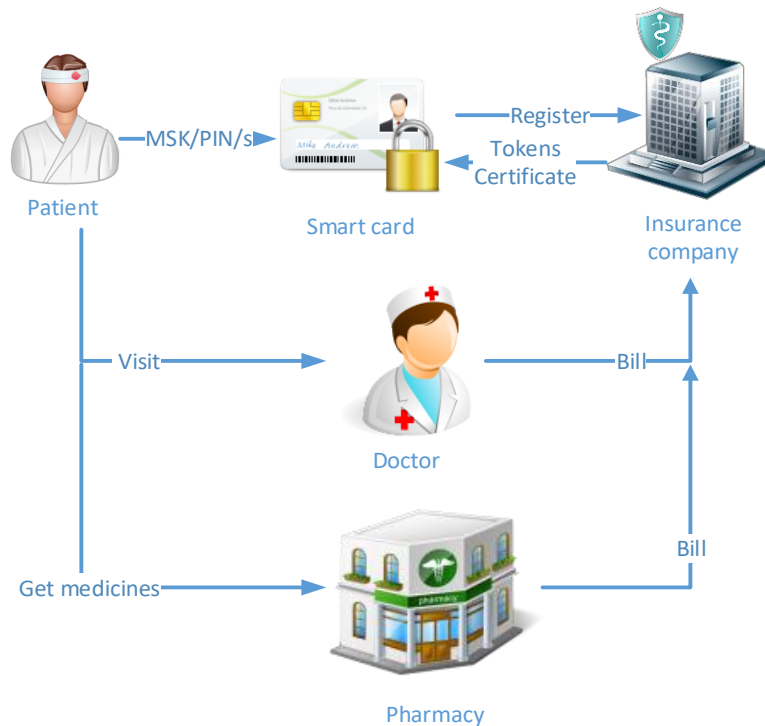


Figure 11: A Simple model of billing procedure with smart card

As shown in Figure 11, a patient firstly registers at an insurance company. In the registration, the patient needs to provide the real identities and some other information to the insurance company, e.g. the employer information as introduced in Section 3.1.3 and 3.3.2.4. After the registration, the patient gets an initialized smart card for the proof of the patient's health insurance. The smart card can be used in many normal healthcare activities as introduced in Section 6.1, e.g., storing MSK and visiting doctors. In order to be used in the billing procedure, the smart card needs to store the publicly known tokens (company name, billing address) of the insurance company. The smart card also stores a certificate issued by the insurance company to the patient. Along with the certificate, a secret private key (SKI) which is only known by the patient is stored in the protected memory of the smart card. On the smart card, a photo of the patient is also printed. Optionally, a description of the insurance coverage can be enclosed. The description must be signed by the insurance company. We have required the smart card to store a master secret key (MSK) generated and only known by the patient in Section 6.1. MSK is stored in the protected memory of

the smart card and used for generating the pseudonyms of the patient. The smart card is also set a PIN (only known to the patient) for authenticating the patient to avoid abuse. In the billing procedure, similar to x in MSK, another secret key “ s ” will be used for signing the bills, which will be introduced in Section 6.2.2.

As we have introduced in Section 6.1.1, each healthcare provider (e.g. doctor or pharmacist) needs to get a certificate ($Cert_h$) from a trusted authority with self-known private key (HSK) which is not depicted in the Figure 11 for simplicity. The digital certificate ($Cert_h$) includes the information like healthcare categories and identities of healthcare providers. The private key HSK needs to be stored in the protected memory of a smart card issued to the healthcare provider with an authenticating PIN. $Cert_h$ and HSK will be used to sign the bills sent to insurance companies, which will be introduced in Section 6.2.2.

6.2.1.2 Validating Patients

As shown in Figure 11, a patient visits a healthcare provider (e.g. a doctor or pharmacy) taking along the smart card described in above Section. The healthcare provider can check the validation of the patient through checking the smart card. Firstly, the photo on the smart card and the PIN required by the smart card are the preliminary checks by the healthcare provider. Further, the healthcare provider can validate the patient’s certificate stored in the smart card and challenge the corresponding SKI to confirm that the patient is truly insured by the claimed insurance company. Optionally, the healthcare provider can check the description of the insurance coverage of the patient if available. Although it is not necessary, an online certificate service provided by the insurance company would help the healthcare provider to validate patients. If the insurance company broadcasts revoked certificate list online regularly, the healthcare provider can check whether the certificate of the patient is expired or in the list of revoked certificates or not in time, and even further inquire the patient’s insurance coverage online (by anonymously sending the patient’s contract reference number to the insurance company). In the process of validating the patient, the healthcare provider does not disclose the identity of the patient to the insurance company. If the healthcare provider is assumed to be trustworthy to protect the privacy of the patient in the process of validation, the insurance company is unable to know the identity of the patient who is visiting the doctor.

6.2.1.3 Generating a Bill

A bill sent from a healthcare provider to the insurance company must enclose all items of the services or drugs that a patient has taken. The list of the items is the first basic information that the bill should contain. Additionally, the healthcare provider’s tokens (e.g., the bank account information, name, address) should also be included. Another important part in the bill is the signatures of the patient (Sig_p) and the healthcare provider (Sig_h). The patient’s signature includes two parts. The first part is the traditional digital signature by using the secret private key SKI corresponding to the patient’s certificate issued by the insurance company as introduced in 6.2.1.1. As an exception, the patient’s certificate is not enclosed in the signature, in order to prevent the insurance company from knowing the identity of the patient directly by the certificate. Thus, the patient’s traditional signature is simply the encrypted (e.g. by RSA with SKI) digest (computed from a hash function) of the data to be signed. However, the healthcare provider must store both the certificate and the signature of the patient in a local storage (e.g. the computer in doctor’s practice), for the sake of

fraud detection as introduced in Section 6.2.1.4. The other part of the patient's signature is the signature by the secret s corresponding to the pseudonym scheme which will be detailed in Section 6.2.2.3. The traditional digital signature of the healthcare provider by using HSK and the certificate (Cert_h) are the last necessary segments of the bill.

From the explanation in above paragraph, a bill is created by following Algorithm 9 where "||" means concatenation.

Algorithm 9: Generating a bill

$B = \text{Services/Drugs List} \parallel \text{Tokens of healthcare provider}$
 $\text{Sig}_p = \mathbf{Sign}_{\text{SKI}}(B) \parallel \mathbf{Sign}_s(B)$
 $\text{Sig}_h = \mathbf{Sign}_{\text{HSK}}(B \parallel \text{sig}_p)$
 $\text{Bill} = B \parallel \text{Sig}_p \parallel \text{Cert}_h \parallel \text{Sig}_h$

6.2.1.4 Validating a Bill

After the insurance company receives a bill from a healthcare provider, the first task is to validate the bill. The validation aims to confirm that the bill is sent from a valid healthcare provider, and that the bill originates from a customer of this insurance company. By checking the validation of the certificate (Cert_h) and the signature (Sig_h) of the healthcare provider, the insurance company can easily verify whether or not the bill is sent from a valid healthcare provider. Because the patient's certificate is not enclosed in the bill as introduced in Section 6.2.1.3, the insurance is unable to check the patient's traditional signature ($\text{Sign}_{\text{SKI}}(B)$) directly in order to verify that the bill originates from a registered customer. Instead, the insurance company needs to check the patient's signature ($\text{Sign}_s(B)$) corresponding pseudonym which will be introduced in 6.2.2.4. The validation of $\text{Sign}_s(B)$ proves that the signature does originate from a customer that is insured by the insurance company, while the validation process does not disclose any information about whom the patient is exactly. For more details about patient's signature corresponding pseudonym, please refer to Section 6.2.2. After succeeding in all the validations on the bill, the insurance company pays to the healthcare provider according to the tokens of the healthcare provider enclosed in the bill.

Because the certificate of the patient or any other information about the patient's identity (e.g. contract reference number) is not enclosed in the bill, the insurance company is not able to discover the identity of the patient directly from the bill. Moreover, the validation of the patient's signature $\text{Sign}_s(B)$ will not disclose any information about the patient's identity as introduced in Section 6.2.2.4. Thus, the identity information of patients is completely protected from insurance companies during the billing procedure, i.e., it decreases the risk that the privacy of patients is disclosed due to insurance companies. Nevertheless, insurance companies can still validate the bills and run their business normally without the knowledge of patients' identities in the billing procedure, which will be introduced in Section 6.2.2.

6.2.2 Pseudonymizing Bills

6.2.2.1 Initialization

When a patient registers at an insurance company, the patient stores in his smart card some numbers $[g, p, q]$ (these numbers are parts of the tokens of the insurance company as introduced in Section 6.2.1.1) generated by the insurance company as follows. The insurance company chooses a k -bits (k is usually no less than 160 bits) prime integer q , and another prime number p ($p > q$) which satisfy $q | (p-1)$. By Z_p^* we denote a multiplicative group modulo p . The insurance company finds $g \in Z_p^*$, to be of order q modulo p . Then g is the generator of the cyclic subgroup G_q . The insurance company stores in the patient's smart card the numbers $[g, p, q]$. The insurance company just needs to generate these numbers only once, because all patients insured by this company will have the same numbers $[g, p, q]$ in their smart cards.

After getting the numbers $[g, p, q]$, the patient chooses randomly a number $s < q$ which is stored into the protected memory of the smart card. s is chosen and only known by the patient. The process of choosing s can be optionally done at the patient's own device or a trusted family doctor's computer.

With the similar theory in Section 5.2, given $g, h \in G_q$, such that h is selected from G_q uniformly at random, it is hard to compute an integer s such that

$$g^s = h \pmod{p}.$$

6.2.2.2 Generation of Pseudonym

Each patient computes her/his initial pseudonym $PIDI = g^s \pmod{p}$. For clarification, $PIDI$ is independent with the pseudonyms that we have introduced in Section 6.1. The computation of $PIDI$ is done at the patient's own device or a trusted doctor's computer (e.g. a trusted family doctor). Then, the $PIDI$ of the patient is sent to the insurance company when the patient firstly uses his smart card at a doctor. The doctor sends a message (MSG) including the patient's pseudonym $PIDI$ to the insurance company. MSG further includes the doctor's signature on $PIDI$ by using the doctor's secret key HSK and certificate $Cert_h$. So the MSG sent from the doctor to the insurance company is defined as follows,

$$MSG = PIDI || \mathbf{Sign}_{\text{HSK}}(PIDI) || Cert_h.$$

Before sending the message, the doctor needs to validate the patient to confirm that the patient is insured by the claimed insurance company as introduced in Section 6.2.1.2. The validation procedure does not disclose any identity information of the patient to the insurance company. Moreover, MSG does not contain any identity information about the patient either. Thus, the insurance company is not able to know whom the pseudonym $PIDI$ belongs to. However, by validating the certificate and the signature of the doctor on $PIDI$, the insurance company can confirm that the MSG does come from a doctor with the certificate $Cert_h$. Doctors are usually assumed as trustworthy in most processes in eHealth systems, although there exist rare scenarios that a doctor might do fraud. A corresponding countermeasure against the malicious behavior of doctors will be introduced in Section 6.2.3.1.

6.2.2.3 Generation of Pseudonymous Signature

A patient signs the basic information B in a bill (refer to Section 6.2.1.3) to generate the pseudonymous signature by using the following Algorithm 10.

Algorithm 10: Patient signs the bill with pseudonym

$H = \text{Hash}(B)$
 Patient chooses a random number r in G_q
 $y = r + H \cdot s \pmod q$
 $\sigma = [g^r, y, PIDI]$

The output σ is the pseudonymous signature on B of the patient with pseudonym $PIDI$. The signature σ only includes the pseudonym $PIDI$ and some numbers computed from the secret s without the identity information of the patient.

Because the signature is generated by the patient's secret key s which is only known by the patient, nobody else can generate a valid signature except the patient himself. The security of the signature is based on the one-way property of the hash function and the difficulty of discrete logarithm problem. The difficulty of forging the signature of the patient (i.e., to find out s) is equal to solve the discrete logarithm problem which is unfeasible in current computation condition. This guarantees that attackers are unable to steal any patient's identity (e.g. contract reference number of health insurance) for malicious usage unless obtaining the secret key s of the patient.

6.2.2.4 Verification of Pseudonymous Signature

Each insurance company stores all initial pseudonyms $PIDIs$ of all the insured customers when MSGs are received from the firstly visited doctors as introduced in Section 6.2.2.2. The $PIDIs$ can be stored in a table of a database operated by the insurance company. Along with $PIDIs$, other information in received MSGs is also stored by the insurance company for further usage. By the way, the insurance company can easily guarantee that each patient chooses different s , because the insurance company centrally maintains $PIDIs$ and thus is able to check collision. If a patient chooses an existing s and $PIDI$, the insurance company can ask the patient to choose a new one.

Once the insurance company receives a bill from a healthcare provider, it extracts out from the bill all the segments as shown in Section 6.2.1.3 and 6.2.2.3, and executes the following Algorithm 11 to verify whether the bill originates from an insured customer by this insurance company or not.

If a customer has submitted his initial pseudonym $PIDI$ to the insurance company by the procedure introduced in Section 6.2.2.2 and signed the bill by his secret key s using Algorithm 10, the insurance company is able to confirm that the bill originates from one of its insured customers when Algorithm 11 returns YES. However, the insurance company is not able to know who the patient is (i.e., the identity of the patient) except that the patient's pseudonym is $PIDI$. Moreover, the insurance company is not able to gather any useful in-

formation to infer the identity of the patient from the bill, because the identity of the patient is never implied in the MSG (refer to Section 6.2.2.2) or the bill (refer to Section 6.2.1.3). Thus, the patient's identity information is completely protected from the insurance company in the billing procedure.

Algorithm 11: Insurance company verify the pseudonymous signature on the bill

```
Check whether  $PIDI$  in the bill exists in the database,  
if not, returns NO.  
else  
   $H' = \text{Hash}(B)$   
  Check  $g^y ? = g^r PIDI^{H'}$ ,  
  if not, return NO,  
  else return YES.
```

The verification procedure of the pseudonymous signature does not involve the randomly generated number r (g^r was already enclosed in the signature σ) in the generation procedure of the signature as introduced in Section 6.2.2.3. Therefore, neither the patient nor the insurance company needs to store the number r .

6.2.3 Potential Threats and Countermeasures

6.2.3.1 Trustworthiness of Healthcare Providers

In the initialization procedure as introduced in Section 6.2.2.1, a patient without necessary knowledge or resources needs to find a trusted doctor for help to set the patient's secret key s and generate the initial pseudonym $PIDI$ of the patient. It is reasonable to assume that the patient is able to find such a trusted doctor to assist with these matters. In practice, a patient usually has a family doctor who is considered as fully trustworthy for keeping the privacy of the patient. Fortunately, some products of smart card have the internal initialization function, e.g., generating a random secret s . Thus, the generation of patients' secret key s and $PIDI$ can be done in smart card automatically. In this case, the initialization module of the smart card must be trustworthy.

In existing eHealth systems, registered healthcare providers holding valid certificates are assumed as trustworthy in keeping the privacy of patients. In practice, the registered healthcare providers are publicly monitored and are obligated by laws to protect the privacy of patients. Nevertheless, in our proposed billing procedure, a dishonest healthcare provider might commit a fraud to get cheated payment from an insurance company. E.g., a corrupt healthcare provider might send a forged message MSG including an invented (nonexistent) patient's initial pseudonym to the insurance company, i.e., the healthcare provider could invent a nonexistent patient with fake secret key s' and $PIDI'$. Thereafter, the healthcare provider could send fake bills to the insurance company on behalf of the invented patient. Responding to this, one solution is that the parameters such as g , p and

q are only known to insurance companies, and they are embedded in the protected memory of the smart cards (even not directly accessible to patients) of patients. As a result, a dishonest healthcare provider is unable to invent such a nonexistent patient without knowing these parameters.

However, the solution in above paragraph (keeping parameters as secrets only known by insurance companies) has risk of disclosing the identity information of patients to insurance companies in the billing procedure. A corrupt insurance company is able to use different parameters for different customers to obtain their identities and trace their healthcare activities. E.g. when two patients A and B register at the insurance company, the insurance company assigns (g_A, p_A, q_A) and (g_B, p_B, q_B) separately to them. Their initial pseudonyms $(PIDI_A = g_A^{s_A} \bmod p_A, PIDI_B = g_B^{s_B} \bmod p_B)$ are received and stored by insurance company. Later, if a bill including pseudonymous signature $\sigma = [g^r, y, PIDI]$ is received by the insurance company, it can try (g_A, p_A, q_A) and (g_B, p_B, q_B) one by one to verify the signature. If (g_A, p_A, q_A) can succeed to verify the signature, then the bill is believed to originate from patient A. Otherwise, the bill originates from patient B.

To solve the rare issues presented in above paragraph, it is necessary that each insurance company has assigned same publicly known parameters to its customers except secret key s . Meanwhile, in order to prevent a dishonest healthcare provider from sending fake bills, we required in each bill sent to the insurance company, the patient signs the bill by SKI using the traditional digital signature scheme (e.g. RSA) without enclosing the certificate of the patient in the signature as introduced in Section 6.2.1.3. The healthcare provider who sent this bill has to save the bill including both the certificate and the traditional signature of the patient at a local storage. Afterwards, a third party trusted by the insurance company is authorized to select some bills randomly to check with healthcare providers. Healthcare providers are required to provide the certificates of the corresponding patients in the bills. With patients' certificates, the third party can verify patients' traditional digital signatures in the bills and finally confirm the bills originate from the insured customers. Because the certificates of patients are issued by insurance companies, any dishonest healthcare providers cannot generate valid certificates and valid signatures. Thus, the fraud behavior of the dishonest healthcare provider could be partially detected. The dishonest healthcare providers have to confront such risk to be detected and punished.

Instead of requiring healthcare providers to store the complete bills (along with patients' certificates) at local storage, an alternative way is to enclose the encrypted patient's certificate into each bill sent to the insurance company. The patient's certificate is encrypted by the public key in the healthcare provider' certificate $Cert_h$. That is,

$$Sig_p = \mathbf{Sign}_{SKI}(B) \parallel \mathbf{Sign}_s(B) \parallel \mathbf{Enc}_{PKH}(Cert_p),$$

where $Cert_p$ is the certificate of the patient, and PKH is the public key in the healthcare provider' certificate $Cert_h$.

Once the healthcare provider is asked by a third party to verify a bill, the healthcare provider decrypts $\mathbf{Enc}_{PKH}(Cert_p)$ in the bill by using the corresponding secret key HSK . The third party can finally validate the traditional digital signature of the patient with the decrypted certificate $Cert_p$ to confirm the origin of the bill.

The above solution has one disadvantage, if a patient's certificate used to sign the bills is known to the insurance company during the fraud detection, the insurance company may map the *PIDI* of the patient to the certificate (real identity) forever. That is to say, if the patient next time signs a bill with *PIDI*, the insurance company can easily trace the patient and know the illnesses of the patient. The patient can certainly change *PIDI* to avoid to be traced in the future. Instead of enclosing the patient's traditional signature $\text{Sign}_{\text{SKI}}(\text{B})$ in the bill, the insurance company has some other options to detect fraud from corrupted doctors. For example, if a corrupted doctor sends a forged MSG to the insurance company, the *PIDI* enclosed in the MSG will be later used to generate fake bills. Thus, the insurance company can do statistic on such behavior and detect high doubtful fraud.

6.2.3.2 Adapting to Practical Billing Models

In practice, different patients may be covered with different ranges by the insurance company. As a result, before a healthcare provider sends a bill to the insurance company, the healthcare provider must check whether the expenses in the bill are covered by the patient's health insurance or not. If not, the expenses that are not covered should be paid by the patient. One simple solution is to store a statement of the coverage range provided and signed by the insurance company in the patient's smart card as we introduced in Section 6.2.1.2. Another solution is to enable the healthcare provider to inquire the insurance range of the patient online from the insurance company by using secure multi-party computation (Yao, 1982). Secure multi-party computation is a popular technology to protect the participants' privacy in a system without a trusted third party. It enables multiple participants to compute the value of a function with their own private inputs, while protecting their private inputs from being known by other participants. The secure multi-party computation can enable healthcare providers to query the insurance coverage range anonymously from the database of insurance companies. Insurance companies do not know what healthcare providers have queried, but healthcare providers can get desired answers from such anonymous queries.

In many practical billing systems, healthcare providers and insurance companies may use bill collectors as agents to deal with the billing. To be adapted in this case, healthcare providers and insurance companies can issue sub-certificates to the bill collectors who execute our proposed solution on behalf of them.

In the billing model with private health insurance companies, a bill from a healthcare provider is firstly sent to a patient and the patient pays the bill to the healthcare provider. Later the patient sends the bill to the insurance company to get reimbursed. In this case, the patient needs to add the personal tokens (e.g. bank account information) into the bill and sign it using the secret key s and initial pseudonym *PIDI*. It will be difficult to protect the patient's identity against the insurance company because the personal tokens somehow disclose the patient's identity. One countermeasure is that, the patient's tokens in the bill are anonymous and the patient can send the bill to the insurance company and get reimbursed in an anonymous manner (e.g. sending bills from an anonymous email address or a trusted agent and get reimbursed through anonymous payment provided by a trusted financial organization).

6.2.3.3 The loss of smart card

Because a patient's secret keys (MSK, s , SKI) are stored in the protected memory of the smart card which is protected by a PIN, a person who obtains the patient's smart card is unable to gain any private information of the patient or get any illegal benefits. The smart card is useless to the person if we assume the security protection in the smart card is trustworthy.

The important matter upon loss of the smart card is to recover its uses in normal healthcare activities, because the secret keys and other information stored in the smart card are lost. When a patient's smart card is lost, the patient must apply a new smart card with new certificate and SKI from the insurance company. At the same time the insurance company will revoke the old certificate in the lost smart card. The patient then generates a new initial pseudonym $PIDI$ (he may use the old s or a new s) and goes to a trusted doctor to register the new initial pseudonym with the insurance company as introduced in Section 6.2.2. The patient can then use the new smart card in a normal way.

A patient's secret MSK must have a secure backup in the custody of each patient in case the smart card is lost, because MSK is only known by the patient, nobody else can recover it. Once the patient lost his smart card, after getting a new smart card re-initialized by the insurance company, he restores the backup MSK into the new card. Thereafter, the EHR entries stored on the cloud (they are created by the old card) can be reused by the new card. The order number $last$ of PID_{last} can be recovered by the patient as follows. The patient reproduces his pseudonyms one by one using Algorithm 4 or Algorithm 5 and sends each one to the cloud to inquire the corresponding EHR entry. Until the cloud returns that the corresponding EHR entry does not exist, the patient stops running Algorithm 4 or Algorithm 5 and gets the serial number $last$ of the last pseudonym that lost smart card has ever produced. Then $last$ has also to be restored into the new smart card. The feasibility of having all patients back up binary secrets depends on the practical condition. Some optional solutions are available. E.g., patients can use public storage service (e.g. cloud disc service) to store their secrets; the general doctors of patients can also help to back up their secrets. The secrets stored at the public storage or general doctors' computers must be protected by patients-only known passwords.

In some rare cases, the smart card and even the secret keys MSK, s and PIN of a patient might be stolen by an attacker. The patient needs to report this to the insurance company in order to revoke the certificates as soon as possible to avoid the abuse of the lost smart card.

6.3 Application in Secondary Use of Health Data

6.3.1 Setting of Secondary Use in Cloud-based eHealth Systems

6.3.1.1 Secondary Users

In Figure 12, we render the procedure of secondary use of health data in a cloud-based eHealth system as introduced in Section 6.1.1. Secondary users are also connected to the cloud to conduct their uses of EHR in the cloud environment.

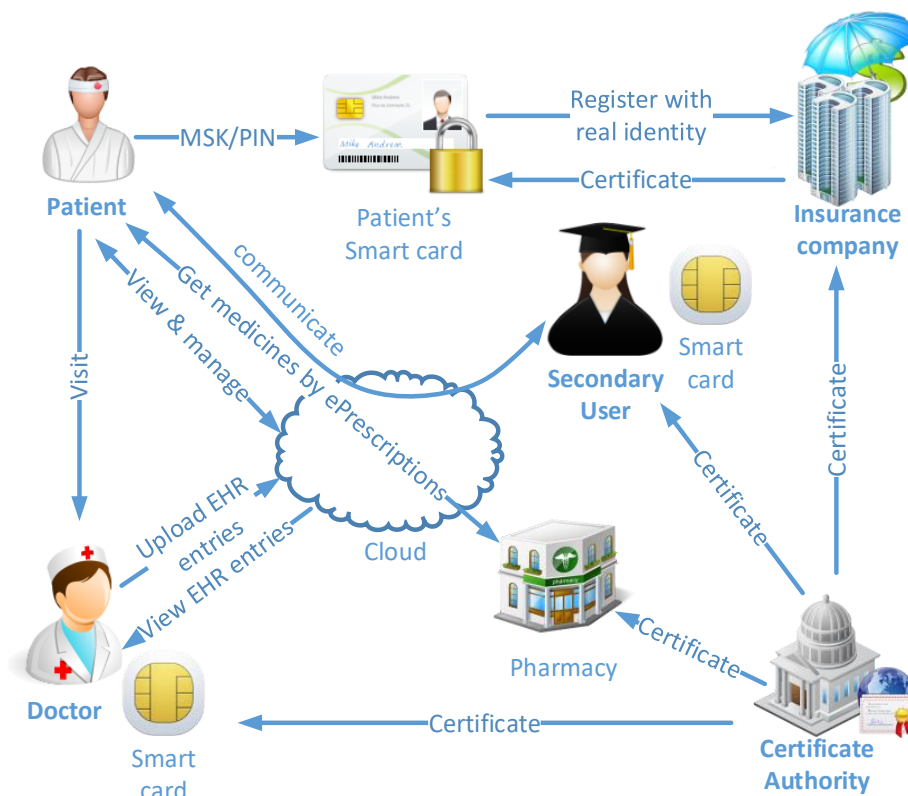


Figure 12: A cloud-based eHealth system enabling ordinary healthcare activities and secondary use

Unlike existing secondary use solutions, secondary users in this thesis do not get EHR from healthcare providers or any trusted third parties. Instead, they obtain EHR from the cloud directly. Moreover, they are able to communicate with patients who are the owners of EHR, to get consent and send feedback.

As a consumption in this thesis, secondary users must be certificated, i.e., the identities of secondary users must be publicly verifiable. To this end, we introduce in our model an entity named “certificate authority”. It issues digital certificates to other entities. Like doctors, pharmacists, and insurance companies, secondary users need to apply certificates from the certificate authority. Each secondary user needs to generate a self-known private key and a corresponding publicly known public key. The public key together with some necessary

information of the secondary user (e.g. name, certificate type and valid period) is signed by the certificate authority to form a digital file i.e. the certificate. The certificate authority could be the health department of the government or an international trusted organization which confirms the qualifications of doctors, pharmacists, insurance companies and secondary users, and issues them digital certificates. The certificates and the corresponding private keys of the applicers can be stored in the protected memory of smart cards owned by them with authenticating PINs to avoid abuse of the smart cards. We render the smart cards of a doctor and a secondary user as examples in Figure 12.

6.3.1.2 Patient's Secrets and Certificate

Each patient generates a major secret key (MSK) which is only known by the patient himself. MSK is very important to the patient, because all the private information in EHR and the private communications with other entities associated to the patient will be protected by MSK. In an eHealth system where smart cards are widely used, the patient's MSK is usually stored in the protected memory of a smart card owned by the patient as shown in Figure 12. A PIN set by the patient is employed to authenticate the patient to avoid abuse.

Unlike the certificates of doctors, pharmacists, insurance companies and secondary users, a patient's certificate is not issued by the certificate authority but by the insurance company in registration procedure as introduced in Section 6.1.1. Each patient generates a pair of private and public key (SKI, IPK), and the public key IPK will be sent to the insurance company in order to get a certificate. The corresponding private key SKI which is only known by the patient could be stored in the protected memory of the smart card owned by the patient. In the registration process, the patient's real identity will be known by the insurance company. However, the patient's identity information is not included in the certificate issued by the insurance company. Typically, a certificate number which can only be mapped to the patient's real identity by the insurance company, a token of the insurance company, the public key of the patient, and the valid period are necessary in the patient's certificate. For the details of the registration with insurance company, please refer to Section 6.1.1 and 6.2.2.1.

6.3.2 Pseudonymized EHR from Ordinary Healthcare Activities

A patient's ordinary healthcare activities include visiting doctors, get medicines from pharmacies by prescriptions, viewing and managing his own EHR, and so on. In these activities, EHR entries will be created, retrieved and updated by different entities in the systems. Because the secondary use process is directly based on EHR entries stored in the cloud from these ordinary healthcare activities, it is valuable to review the format and content of EHR entries. In following two Sections, we review how the pseudonyms of patients are generated and used to manage EHR. During the review, we also discuss how to refine the format of EHR to facilitate the secondary use.

6.3.2.1 Visiting a Doctor

A patient visits a doctor, taking along his smart card. The doctor firstly validates the patient by checking the patient's certificate and challenging the patient's SKI with the help of the insurance company's root certificate. As introduced in 6.1.2, after the doctor's diagnosing and treating, the doctor writes down a record (including diagnosis, examinations, treatment

and other information), with a prescription stating the medicines that the patient needs to buy at a pharmacy. Meanwhile, a new pseudonym (PID) and an encryption key (EK) are generated from MSK by the patient's smart card. Both the record and the prescription will be encrypted by EK and signed by the doctor's private key. Afterwards, all the data from the doctor are uploaded to the cloud along with an index header, the new PID, forming a new EHR entry for the patient. Thus, an EHR entry needs to be at least compatible with the following sections where "||" means concatenation,

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

According to the above definition of EHR entry, a patient's EHR contains many EHR entries as the patient visited doctors many times. Each EHR entry has a unique pseudonym PID and a separate encryption key EK.

The cloud stores EHR entries of each patient by indexing their PID section. The cloud can check the validity of doctors by verifying their certificates and challenging their private keys with the trusted authority's help to avoid illegal uploading of EHR entries as introduced in 6.1.3. The PIDs are regenerated by patients when EHR entries need to be retrieved. For example, when a doctor wants to view the anamnesis of a patient or a patient wants to view his own EHR, the patient regenerates all the PIDs one by one and sends them to the cloud in order to retrieve the content in corresponding EHR entries.

Due to the use of pseudonyms, the content in an EHR entry could actually be partially encrypted. For example, the diagnosis, examinations and even prescription in one EHR entry could be plaintext. However, some sensitive information such as age, gender, DNA information and other identifiers are encrypted by EK, because this information could be used to deduce the real identity of a patient. A partial encrypted EHR entry looks like as follows,

Partial encrypted EHR entry = PID || Enc_{EK} (sensitive identifiers) || record
|| Sign_d (record) || Enc_{EK} (prescription) || Sign_d (prescription).

The partial encryption of EHR entries will benefit secondary use as introduced in Section 6.3.3.2.

6.3.2.2 Getting Medicines from a Pharmacy

A patient goes to a pharmacy to buy medicines by using a prescription which is enclosed in one of his EHR entries. The patient retrieves the EHR entry from the cloud by providing the cloud a PID (the EHR entry with this PID has an unused prescription) and shows the decrypted prescription with the doctor's signature to the pharmacist. The pharmacist can validate the prescription by checking the signature of the doctor. If the signature is valid, the pharmacist generates an additional signature on the prescription indicating that the prescription has been used and asks the patient to update the original prescription's signature section of EHR entry by including the pharmacist's signature. The cloud needs to check beforehand whether the patient is the owner of EHR entry or not by authenticating PID. For more details about the process in pharmacy, please refer to Section 6.1.5. Due to the updating of the prescription's signature, the updated EHR entry with a pharmacist's signature looks like this,

updated EHR entry = PID || Enc_{CEK} (sensitive identifiers) || record
 || Sign_d (record) || Enc_{CEK} (prescription) || Sign_d (prescription) || Sign_p (prescription).

6.3.3 Secondary Use of Pseudonymized EHR

Although all patients' EHR stored together in the cloud can be publicly accessed (even in plaintext due to partial encryption) by any secondary user directly, a secondary user needs more information from patients for further use of EHR. E.g., to get consent from patients, to obtain encrypted identifiers like ages and genders.

Formally, the process of secondary use is as follows: the secondary user sends an invitation to a patient by marking one EHR entry of the patient; the patient then responds to the secondary user by sending the secondary user a reply; the secondary user sends feedback to patients if available. The detailed process of secondary use of the pseudonymized EHR will be presented in the following Sections based on the pseudonym solution presented in chapter 5. In the process, the pseudonyms will play an important role to protect the patient's real identity against the secondary user in the communications. We emphasize the requirements to such a pseudonym scheme in Section 6.3.3.1.

6.3.3.1 Requirements to the Pseudonyms

- Decentralized, unique and independent

This requires that the pseudonyms cannot be generated by others without knowing the patient's major secret key MSK. Only the patient can generate a new PID for each EHR entry. The PIDs should be independent from each other and collision free (unique).

- Irreversible

One or more pseudonyms of one patient cannot be used to deduce the real identity of the patient. An attacker without knowing MSK cannot find out other pseudonyms of one patient from the knowledge of some PIDs of the patient.

- Ownership proof

Pseudonyms can be used to verify the patient's ownership on one EHR entry without revealing the patient's real identity to the verifiers.

Such a pseudonym scheme proposed in chapter 5 protects the privacy of patients and security of EHR stored in the cloud when EHR are used in both ordinary healthcare activities and secondary use.

Besides a proper pseudonym scheme, some corresponding protocols to utilize the pseudonyms are also important to prevent the disclosure of the patient's privacy. Especially in the secondary use, because secondary users need to contact patients to get consent from and send feedback to patients, communication protocols based on the pseudonym scheme should be carefully designed to avoid potential disclosure of patients' private information.

6.3.3.2 Searching for Participants

As introduced in Section 6.3.1 and 6.3.2, the pseudonymized EHR entries of all patients are publicly accessible in the cloud. A secondary user such as a clinical researcher on cancer project is able to surf in the cloud to look for target patients as participants. Due to the partial encryption of EHR as introduced in 6.3.2.1, the researcher can use keywords such as “cancer OR tumor” for fast search in the record segment. Because the record segment is not encrypted, the cloud can provide powerful search service based on plaintext. The cloud returns to the researcher all available EHR entries which match the key words.

The researcher finds out the target EHR entries, but they are owned by unknown patients. The researcher usually needs more information for his research. Firstly, the researcher needs to know which of these EHR entries belong to one patient. A patient may have multiple separated EHR entries regarding one disease, e.g., a chronic disease needs to be treated for many times, and each time a new EHR entry with a new PID is created. Because the PIDs of these EHR entries are independent from each other without knowing the secret key as introduced in Section 6.3.3.1, the researcher is unable to concentrate the EHR entries of the same patient. Secondly, the researcher may want to obtain some further health information (e.g. related diseases) about patients. However, the researcher is unable to find out any other EHR entries of a patient, because the researcher has no way to know what other pseudonyms the patient has. At last, the researcher may want to know some personal identifiers of patients (e.g., age and gender) and even some more information which is encrypted in EHR entries by the encryption key EK only known by patients. As a result, the researcher has to be able to contact these target patients for help because the desired information can only be provided by the owners who knows MSK.

6.3.3.3 Contacting Target Patients for Consent and More

Because EHR are highly private information, using these data outside of clinics or hospitals is usually restricted in laws and regulations. A researcher is obliged to get signed consent from patients to use their EHR for some special purposes (e.g. medical research) other than ordinary health care. Moreover, the researcher may be eager to know more information about patients as introduced in Section 6.3.3.2.

After finding out in the cloud an EHR entry which may come from a potential participant, the researcher prepares a digital file (denoted as invitation) to ask for the patient’s consent. The invitation must declare the purpose and other concerned information about the research. Moreover, the invitation can also include what information the researcher needs more (e.g. gender, age, other EHR entries for related diseases) from the patient.

Then the researcher sends the invitation to the patient who owns the EHR entry by asking the cloud to mark that found EHR entry as follows,

Marked EHR entry = Original EHR entry || Invitation || Sign_R (Invitation),
where Invitation = uid || Cert_R || Consent.

The content of “Original EHR entry” was introduced in Section 6.3.2.1 and 6.3.2.2. Cert_R is the certificate of the researcher, which was introduced in Section 6.3.1.1. Sign_R (Invitation) means the signature of the researcher on the “Invitation”. In the “Invitation”, uid is a long

serial number (collision free) that the researcher generates intending to assign a unique number to each patient. Because the researcher does not know which EHR entries come from a same patient, he just labels each different EHR entry with a different uid. We use “Consent” to denote all the information which the researcher wants to inform the patient (e.g. purpose of the research) and ask for from the patient (e.g., age, other EHR entries on related diseases). “Consent” may look like a questionnaire.

Before accepting the researcher to mark EHR entries, the cloud must validate the researcher by checking the certificate and challenging the private key of the researcher. We assume that the cloud is semi-trusted and has the structure knowledge of EHR entries as we have discussed in Section 6.1.7.5. Some more discussion about the trustworthiness of the cloud in secondary use is presented in 6.3.4.3.

6.3.3.4 Responding to the Secondary User

The patient who is the owner of the “Marked EHR entry” gets a notification when he accesses his EHR entries in the cloud afterwards. All information enclosed in the invitation can be directly read by the patient. The patient has right to make own decision whether participates or not. Before accepting the invitation, the patient can at first check the validation of the “Invitation”. He can check the researcher’s certificate and the signature of the “Invitation”. To accept the invitation, he can update the “Marked EHR entry” by attaching a “Reply” to the marked EHR entry as follows,

$$\begin{aligned} \text{Agreed EHR entry} &= \text{Marked EHR entry} \parallel \text{Reply}, \\ \text{where Reply} &= \text{Enc}_{\text{PKR}}(\text{uid} \parallel \text{Filled Consent} \parallel \text{Sign}_p(\text{Filled Consent}) \parallel \text{EK}). \end{aligned}$$

Enc_{PKR} means encrypting by the public key of the researcher in the researcher’s certificate. The patient fills the requested information, e.g., the identifiers like age and gender, into the consent to form the “Filled Consent”. Sign_p means that the patient signs the “Filled Consent” to generate a signature. EK is the encryption key of the corresponding EHR entry as explained in Section 6.3.2.1. If a patient has only one EHR entry marked by the researcher, he just encloses the uid provided by the marked EHR entry into the reply. If a patient has multiple EHR entries marked by the researcher, each EHR entry was assigned with a different uid. The patient has to choose an arbitrary one from the assigned uids and enclose it into all his agreed EHR entries. If the researcher required the patient to provide other EHR entries of some related diseases, the patient can attach the corresponding $\text{Enc}_{\text{PKR}}(\text{uid} \parallel \text{EK})$ to the end of other related EHR entries which were not marked by the researcher. Meanwhile, the patient needs to enclosed the PIDs of these related EHR entries to the filled consent.

In the reply, the patient has to tell the researcher EK to enable researcher to decrypt the required information which is encrypted in an EHR entry. If the EK was used to encrypt the patient’s private information which is not intended to be known by the researcher, the patient can derive a new EK’ from EK (e.g. $\text{EK}' = \text{Hash}(\text{EK}+1)$) to encrypt the information that can be known by the researcher. Instead of EK, EK’ is enclosed in the “Agreed EHR entry”.

Before the cloud accepts the patient’s reply which needs updating on an EHR entry, the cloud must verify whether the patient is the owner of the EHR entry or not in order to avoid

malicious updates. The algorithms in Section 5.5 can be used to do the verification without disclosing any identity information of the patient to the cloud.

The patient can also decline the invitation from the secondary user by saying “No” in the “Reply” or simply removing the secondary user’s “Invitation” or giving no “Reply” on the “Marked EHR entry”.

The patient can use the certificate from the insurance company and the corresponding private key (SKI) to sign the “Filled Consent”. However, sometimes the patient may mind the secondary user’s knowing his certificate from the insurance company, because a secondary user might collude with the insurance company to discover the patient’s real identity and private information in his EHR entries. Responding to this case, the patient can apply another certificate at the certificate authority in order to communicate with secondary users. It could be a kind of anonymous certificate where the patient does not need to provide the real identity to the certificate authority. However, the anonymous certificate and signature enable an attacker to impersonate the patient to reply a secondary user, because the attacker can use an arbitrary certificate applied from the certificate authority. The attack can be ruled out due to the following two reasons. Firstly, the attacker can not pass through the ownership verification challenged by the cloud when the attacker tries to attach the forged reply at the end of a marked EHR entry. The cloud will not allow the attacker to add forged “Reply” if the attacker fails to prove the ownership of the EHR entry by the algorithms in Section 5.5. Secondly, even if the attacker succeeds in adding a forged reply to a marked EHR entry, the secondary user can easily drop the forged reply by checking the encryption key EK. Because the attacker does not know the correct EK, the secondary user can distinguish the forged reply from a failure of decrypting by incorrect EK. Moreover, if the patient has multiple EHR entries marked by the secondary user, because the attacker cannot know which marked EHR entries belong to the same patient, the secondary user can check the consistence of the information (e.g. birth date) in these multiple EHR entries to drop forged replies.

Besides the anonymous certificate and signature, there is another solution if EK is mandatory in “Reply”, i.e., the patient needs to provide EK to the secondary researcher. Because EK is only known by patients, EK can be used to “sign” the reply to prove the reply does come from the owner who knows EK. The patient uses EK to generate a digest by a keyed hash function. $\text{Sign}_p(\text{Filled Consent}) = \text{KHash}(\text{Filled Consent}, \text{EK})$. After the researcher extracts EK from the Reply, the researcher can check the digest by the extracted EK. Because the EK is only known by the patient, the digest can only be generated by the patient. Thus the digest can be seen as a signature from the patient.

6.3.3.5 Sending Feedback to Patients

The researcher gets a notification if a marked EHR entry was replied by the owner. The received “Reply” in the agreed EHR entry can be decrypted by the researcher’s private key. Then the filled consent and the signature of the patient can be extracted out and validated. Afterwards, the encrypted data (e.g. gender and age) in the EHR entry can be decrypted by using the EK enclosed in the reply.

The researcher distinguishes different patients by uid, i.e., all “Agreed EHR entries” with the same uid come from one patient. From the filled consent, the researcher could find out all

necessary information to conduct the research. After the researcher finishes the research project, there may be some feedback which interests the participating patients. The feedback can be sent to a patient by the researcher through updating the agreed EHR entry like following,

Feedback EHR entry = Agreed EHR entry || $Enc_{EK}(\text{Feedback} || \text{Sign}_R(\text{Feedback}))$.

The “Feedback” can include any messages sent from the researcher, e.g., the result of the research, advice to the patient’s disease. $\text{Sign}_R(\text{Feedback})$ is the researcher’s signature on the “Feedback”. The “Feedback” and the signature from the researcher are encrypted by the encryption key EK of the corresponding EHR entry. Only the patient who knows the EK can decrypt the feedback. It prevents attackers from discovering any private information in the feedback.

6.3.4 Potential Threats and Countermeasures

6.3.4.1 Partial Encryption

For the sake of secondary use, we suggest to use partial encryption on EHR entries as we introduced in Section 6.3.2 and 6.3.3.2. Actually, partial encryption can also benefit the ordinary health care activities. For example, a doctor can discover desired information in a patient’s EHR entries more efficiently. E.g., if the doctor wants to know the allergic history of the patient, he can use keyword “allergy” to search in all EHR entries with the powerful searching service from the cloud, instead of searching in all downloaded and decrypted EHR entries.

Partial encryption has no harm in theory to the privacy of patients if our proposed pseudonym solution is applied. However, in practice, some contents in EHR entries may contain identifiers not easily discoverable. E.g. the name of the patient in the X-ray picture. Thus, the risk of partial encryption depends on the degree of removing all sensitive identifiers of patients and encrypting these identifiers by EK.

In the procedure of secondary use, a secondary user has chance to obtain these encrypted identifiers through the communication with patients as introduced in Section 6.3.3.3 and 6.3.3.4. The secondary user may request some sensitive identifiers from patients, and be able to deduce the real identities of some patients by using advanced analysis technologies introduced in Section 3.2.5. Regarding to this issue, there should be a trusted party, e.g. the organization who issues certificates to secondary users, to evaluate the information enclosed in the invitation sent to patients. Only necessary information is allowed to request from patients.

6.3.4.2 Decision on Own Risk

Even if there is a trusted party to evaluate the information enclosed in the invitation sent to patients as introduced in Section 6.3.4.1, patients must rely themselves to decide whether to participate in the secondary use or not. This may require that patients have some basic knowledge on medicine and secondary use. Especially, patients may be asked to provide some other EHR entries on related diseases. It may impose risk to the privacy of the par-

icipating patients. Although a patient may decline the invitation easily, it might cause deviations and even errors to the results of the secondary use if too many patients say no. One practical countermeasure is to involve doctors, who can give professional suggestion to patients, into the procedure of secondary use to help patients. A communication channel between patients and doctors can be set up by the similar solution in Section 6.3 to discuss the questions on the details of secondary use.

6.3.4.3 Trustworthiness of Cloud

In this thesis, the cloud is not only used as a scalable storage media for EHR, it is also required to perform some validations. E.g., it verifies the ownership of EHR entry when a patient wants to update EHR entry to send a reply in the secondary use, and it also validates a secondary user who wants to send an invitation to an EHR owner by marking one target EHR entry. This may require the cloud be honest or semi-trustworthy. It is reasonable in practice, because a dishonest cloud cannot benefit from the unexpected operations (e.g., modifying or forging the messages in secondary use) to get more private information of patients. It only disturbs the normal process of secondary use. Furthermore, a dishonest operation can be easily detected by patients or secondary users, because all the messages between patients or secondary users have integrity check (e.g., encryption or signature).

6.3.4.4 Other Potential Attacks in Secondary Use

A skilled attacker would succeed in modifying a marked EHR entry by replacing a secondary user's certificate with the attacker's certificate. This attack can be easily ruled out by the patient from validating the certificate and signature of the secondary user before he proceeds to the next step of the secondary use.

Because the communication between patients and secondary users takes place in the cloud, an attacker can easily eavesdrop the messages transmitted over the network or updated to the cloud. However, in our communication protocols, all critical information in the messages is encrypted by the secret keys only known by the communicators. Thus, the eavesdropper cannot get any private information of patients from the communications between patients and secondary users, except the pseudonymized EHR contents existing in the cloud.

7 Implementation

7.1 About the Implementation

7.1.1 Joint Work

This implementation is a collaboration work with Tobias Wilken. He made a deep survey on the appropriate tools and platforms to implement the pseudonym solution and the eHealth system presented in this thesis and our published papers (Xu and Cremers, 2014a, Xu and Cremers, 2014b, Xu et al., 2014). Then he developed the first version of the implementation under Linux. After that, based on the reviews and comments from me and Prof. Dr. Amin B. Cremers, some updates and fixes were made.

I continued to work on the implementation after the graduation of Tobias Wilken. Firstly, the development environment under Windows was set up. Then I revised the source codes to fix some bugs and re-designed some user interfaces. Moreover, I added the full support to the secondary use of EHR.

7.1.2 Development Environment

The web front UI (user interfaces) are written by HTML and JavaScript. The backend server is supported by Python. We also have used a lot of public tools in Python, e.g., Flask, Jsonschema, and Gunicorn. For the detailed development tools, please refer to our source codes.

We have chosen MongoDB as the database software. MongoDB is a cross-platform document-oriented database. Classified as a NoSQL database, it eschews the traditional table-based relational database structure in favor of JSON-like documents with dynamic schemas (MongoDB calls the format BSON), making the integration of data in certain types of applications easier and faster. The instructions of setting up MongoDB are also detailed in the source codes.

The PaaS cloud is supported by cloudControl¹⁴. We deployed the eHealth system on this platform and it can be accessed publicly. We also did a lot of performance tests of our implementation on this cloud. However, the implementation is not depending on the cloud server. It can also be deployed at any servers which are properly configured as described in the documents of our source codes.

¹⁴ <https://www.cloudcontrol.com/>

7.1.3 Availability of the Implementation

The source codes of the application are uploaded at GitHub¹⁵. The current the eHealth system resides at cloudControl¹⁶ which is publicly accessible.

7.2 Architecture of the Implementation

The implementation is divided into three layers as shown in Figure 13: local clients, PaaS applications, and database. The clients can be run at any web browsers with HTML5 and JavaScript support. Patients, doctors, pharmacies and any other entities in the eHealth system use the clients (with user interfaces) to view and operate EHR. The PaaS layer running in the cloud includes a lot of applications which serve different healthcare activities, e.g. visiting doctor, purchasing medicines and secondary use. The Database layer can be distributed MongoDB's in the cloud. They provide storage and access of all EHR.

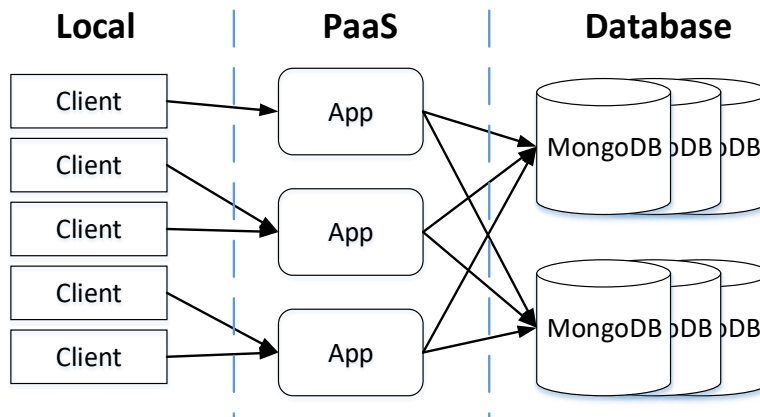


Figure 13: Architecture of the prototypical eHealth system

7.3 User Interfaces

7.3.1 Patient's Secret Key Setup

When a patient uses the eHealth system for the first time, a secret key MSK and PIN are needed to set up according to Section 5.2. In our implementation, the procedure of choosing MSK is automatically done by a JavaScript module executed in the browser that the patient uses. The patient just needs to set a PIN for the smart card as shown in Figure 14.

¹⁵ Source codes available from: <https://github.com/TooAngel/Implementation-PaaS-Reference-Implementation-of-a-pseudonymity-based-eHealth-cloud-system>

¹⁶ Application experience at: <https://prioapbecs.cloudcontrolled.com>

Choose smart card

For this proof of concept the local storage of the browser is used to store the secrets of the patient.

Smart card found. Please provide your PIN.

PIN

Delete

Submit

Figure 14: A patient sets up the PIN and secret key

A patient's smart card is simulated by the Cookies of web browser (which can be seen as the memory of the smart card) and a JavaScript module (which can be seen as the program in the smart card). The patient's MSK and other information will be stored at the local client (i.e. the Cookies of web browser) which are protected by PIN. Different patients are distinguished by PIN if more than one patient logs in the same client (e.g. uses the same PC). A "Delete" button is provided to clear all the smart cards stored. However, the EHR entries related with the deleted smart cards will not be deleted. The secret keys of each patient can be backed up just by coping and saving the corresponding Cookies in the browser. A patient's records can be accessed from another computer where the backup of the patient's secret keys is restored.

7.3.2 Visiting Doctor

The interface provided to doctors for writing records and prescriptions is shown in Figure 15. A doctor asks a patient to input the PIN of the smart card when the "Create" page is accessed.

In this page, a doctor can write anything about patient's illness in the records. For example, the blood sugar of the patient can be input in the left-bottom blank. The doctor can also prescribe to the patient by adding different medicines and the doses in the right side. Both the record and prescription can be chosen as encrypted or not. After writing down all these things, the doctor clicks the "save" button to upload EHR entry to the cloud according to the description in Section 6.1.2.

7.3.3 Viewing and Managing EHR

A patient or any doctor who is treating the patient can view all EHR entries of a patient after the patient inputs the correct PIN and provides the "smart card" (Cookies) at "View" page as shown in Figure 16.

Home **Create** View ePrescription Secondary use

save

This page enables health care providers, especially doctors, to write down the healthcare information (e.g. examinations and diagnosis) about a patient and be used at pharmacies. The Record can contain arbitrary keys and values. Prescriptions contain each medicine's name and an integer indicating the maximum prescriptions can be stored encrypted or unencrypted.

Record

Name:

Provide a name for this value, like examination, bloodsugar and so on.

Value:

Encrypt **Add**

Prescription

Add

Figure 15: A doctor creates an EHR entry for the patient

Home Create **View** ePrescription Secondary use

This view shows all information stored for the patient. The list shows all existing EHREntries. The main part shows a summary of information grouped by a graph, examinations and prescriptions, both shown as a list. This tab can be used by the doctor or patient to get an overview of the health situation and

List of EHREntries

Date	PID
2015-07-02 12:40:09	5edb4105b0
2015-07-02 14:21:58	64e4a4e481
2015-07-02 14:22:47	9ed0d13809
2015-07-02 15:05:00	14ea7df7ae

blood sugar

Date

Examinations

- 2015-07-02 15:05:00 diabetes
- 2015-07-02 14:21:58 Diabetes
- 2015-07-02 12:40:09 a

Prescriptions

Hydrocodone 0 / 10

Figure 16: EHR of the patient is viewed

In Figure 16, all information about the patient's EHR is displayed. On the left side, the pseudonyms of all EHR entries of the patient are listed according to the algorithm in Section 5.4 and the protocol in Section 6.1.4. Each pseudonym can be clicked and the details about the corresponding EHR entry will be popup. The patient can manage EHR entry at the popped page, e.g. deleting it. The cloud will in background check the patient's ownership on the EHR entry to be deleted. Moreover, in the right top of Figure 16, a statistic about the patient's blood sugar is depicted in a chart. In the right bottom, all examinations and prescriptions of the patient are displayed.

7.3.4 Purchasing Medicines

The patient goes to the pharmacy and inputs the PIN for the "smart cards" as introduced in Section 6.1.5, the pharmacist retrieves all the patient's prescriptions and sells the medicines written in them at "ePrescription" page as shown in Figure 17.

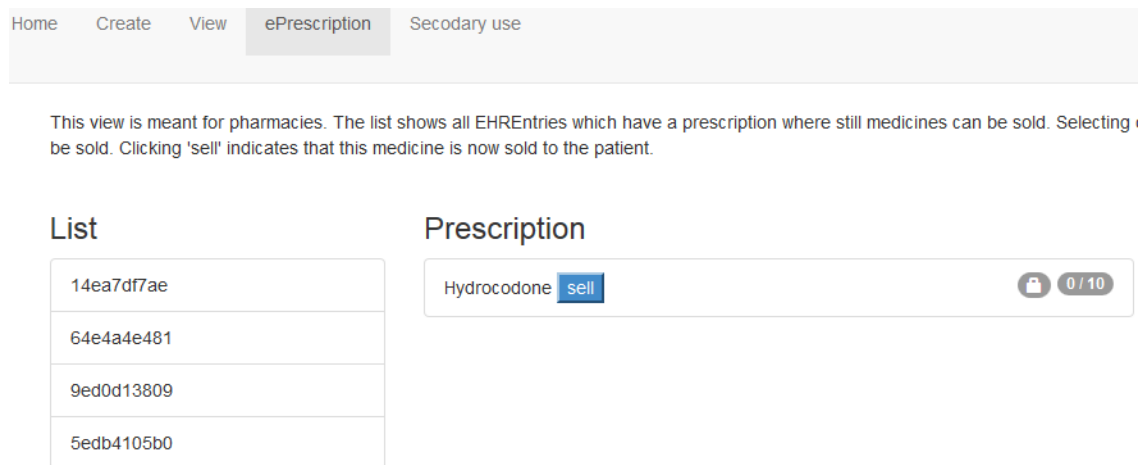


Figure 17: A pharmacy sells the medicines in the prescription to the patient

The pharmacist clicks the "sell" button, which simulates one dose of the medicine is sold to the patient. The cloud will update the EHR entry by adding the pharmacist's signature on the prescription after successfully checking the patient's ownership on the EHR entry.

7.3.5 Secondary Use

A secondary user can view all EHR entries of all patients in the cloud as introduced in Section 6.3.3.2. Some basic searching functions in the unencrypted segments of EHR entries are also possible as shown in Figure 18.

Home Create View ePrescription **Secondary use**

This tab can be used by researchers to search for EHREntries with specific criteria and communicate with them. To filter results that fit to the given criteria, the Query field can be used to select specific EHREntries. The format of the query is: `field operator value` (lower than), `eq` (equal) or `gt` (greater than), the value needs to be a float.

List

7232dcf3ef
5d45af9274
b68ac9a620
dd3a0199c6
17d4fdb74
beb8cab8f9
5edb4105b0
64e4a4e481
9ed0d13809
14ea7df7ae

Query

Provide a query for secondary use

Figure 18: A secondary user views all EHR entries from the cloud

On the left side of Figure 18, the pseudonyms of all available or filtered (by search) EHR entries are listed. The secondary user can click any entry and the detailed content in the corresponding EHR entry is displayed as shown in the top of Figure 19.

14ea7df7ae

Record

bloodsugar 100

EncryptedName EncryptedValue

Prescriptions

Encrypted Medicine 0 / 0

send

Send Request

This form allows researcher to send requests to the owner of the EHREntry, arbitrary key/value pairs are allowed.

Request Add

EK?

age ?

gender ?

name ?


Figure 19: A secondary user sends the requests to the patient

The secondary user can not read the encrypted contents of an EHR entry, as named by “Encrypted*”. If the secondary user wants to request consent, the encryption key (EK) and any other information from the patient, he can select the “Request EK?” checkbox and input these requests in the bottom of the page in Figure 19. At last the secondary user clicks the “send” button to send all requested information (invitation) to the patient as introduced in Section 6.3.3.3.


The patient will receive a notification after the secondary user sent out the invitation as shown in Figure 20. The patient can view all requests from the secondary user, and decide what kinds of information to be replied as introduced in Section 6.3.3.4.

14ea7df7ae

Record

bloodsugar	100
date	2015-07-02 15:05:00
examination	diabetes 

Prescriptions

Hydrocodone	 0 / 10
-------------	--

Request

[Delete](#)

age ?
gender ?
name ?
request_EK yes Send EK? <input checked="" type="checkbox"/>
weight ?

[Reply](#)

Response

age 60
gender male
send_EK Yes
weight 70kg

name	<input type="text" value="I won't tell."/>	add
------	--	---------------------

[Delete Entry](#)

Figure 20: The patient replies to the secondary user

As shown in Figure 20, the patient answered some questions and agreed to send the encryption key (EK) to the secondary user. After the patient clicks the button “reply”, all information will be sent to the secondary user.

The secondary user afterwards can view the unencrypted and encrypted information (as listed in “Record” segment) if the encryption key (EK) was provided by the patient as shown in Figure 21.

14ea7df7ae

Record

bloodsugar 100
date 2015-07-02 15:05:00
examination diabetes

Prescriptions

Hydrocodone	0 / 10
-------------	--------

Notifications

```
Request: {"age":"?","gender":"?","name":"?","request_EK":"yes","weight":"?"}
Response: {"age":"60","gender":"male","name":"I won't tell.,"send_EK":"yes","weight":"70kg"}
```

Send Request

This form allows researcher to send requests to the owner of the EHREntry, arbitrary key/value pairs are allowed.

<input type="text" value="key"/>	<input type="text" value="value"/>	<input type="button" value="Add"/>
----------------------------------	------------------------------------	------------------------------------

Figure 21: The secondary user views the responses from the patient

In this page, the secondary user can view the communication history including all the responses from the patient in “Notification” area. The secondary user can continue to use the page in in Figure 21 to send further feedback to the patient as introduced in Section 6.3.3.5.

7.4 Performance Evaluation

7.4.1 Evaluation on the Cloud Side

In our pseudonym solution and its applications, the cloud just performs as storage and deals with light-load transactions (like authentication, indexing, and searching), The critical operations like encryption and pseudonym generation are not done by the cloud. Instead, they are executed at the clients. Our pseudonym solution does not impose heavy computation pressure to the cloud compared to an ordinary cloud based application, except that a cloud-based eHealth system might have huge data size (EHR entries) and numerous users. The cloud does excel in the big data storage and heavy concurrence with scalable resources.

Having done a lot of tests on the server side to examine the performance of the applications in the cloud, we conclude that the implementation is suitable to be deployed in a public cloud and have good performance in scalability. For more details about these tests, please refer to the load tests in the thesis of Tobias Wilken (Wilken, 2014). Also we will present some test results in the Appendix.

7.4.2 Evaluation on the Client Side

The performance of the clients (e.g. patients) is also expected as sufficient for practical use, based on tests we have done in Section 7.4.2.1 and estimation presented in Section 7.4.2.2. Although we did not test on real smart cards due to the limitation of condition, we have simulated the smart card by computer software and conducted sufficient estimation on the performance in real smart cards as introduced in the following Sections.

7.4.2.1 Performance of Simulated Smart Card by Computer Software

According to Algorithm 2, each pseudonym generation needs two modular exponentiations, each of which (assuming that the length of q is 256 bits) needs about 192 modular multiplications which are most time-consuming. The computation complexity of a modular exponentiation is $O(n)$ (Knuth, 1981) where n is the bit length of exponentiation without pre-computation (Dimitrov et al., 1998). In our implementation, we use JavaScript module to simulate the smart card for generating pseudonyms, encrypting/decrypting and authentication, and we find that the clients (i.e. the browsers) response very fast on the computer with a plain configuration. According to (Jahani et al., 2014), one computation of modular multiplication with 256 bits costs 0.087ms on a test PC (AMD Phenom 9950 Quad-Core processor). So each pseudonym generation nearly costs $0.087 \times 192 \times 2 = 16.704$ (ms). Thus, in our eHealth system, if a patient has 100 EHR entries, the time for generating all pseudonyms of the patient is no more than two seconds.

To examine the software performance of our proposed solution in chapter 5, we implemented and evaluated the Algorithm 1 (setting up secret key MSK) and Algorithm 2 (generating pseudonyms) with computer software. The test software is written in Python 2.7.9 with gmpy2 2.0.3 under windows 10 64bits. The hardware platform is Intel Core i5 520m 2.4GHZ CPU with 8G bytes RAM.

The test result for evaluating Algorithm 1 is shown in Figure 22. In the test, we vary the length of the secret key (MSK) from 128 bits to 256 bits. For each key length, we repeat the initialization for 20 times (i.e., to find out 20 MSKs). The curves in Figure 22 represent the average, maximum and minimum time for initializing a single MSK. From the test, we can see that the worst initialization time is no more than 10 seconds. Because the initialization process only runs once when a patient receives the smart card, a reasonable long time for the initialization is acceptable.

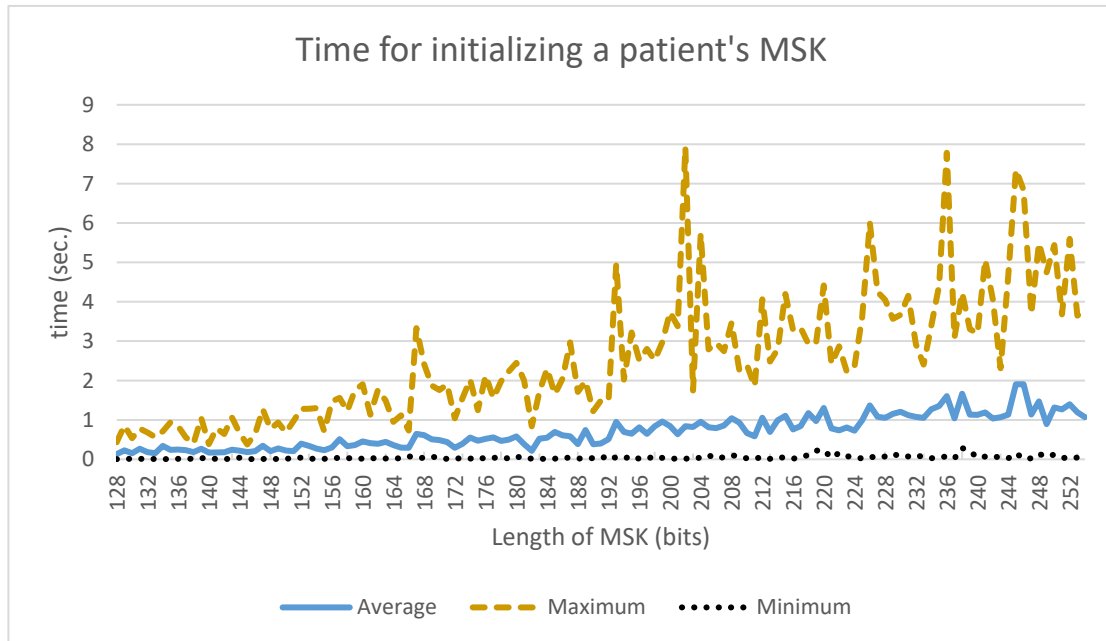


Figure 22: The performance in initializing a patient's MSK with computer software

The test result for evaluating Algorithm 2 is shown in Figure 23. In this test, we have chosen 128 MSKs from 128 bits to 256 bits. For each MSK, we generate 500 pseudonyms. The average time, maximum time and minimum time for generating one pseudonym are depicted in Figure 23. From the test, we can see that the maximum time for generating one pseudonym is less than two milliseconds. Therefore, for a patient with 500 pseudonyms (i.e., 500 EHR entries), the worst time for generating all pseudonyms of the patient is no more than one second.

From the experience in the hardware and software development, the smart cards with special hardware (e.g. ASIC) should have no worse computation performance than an ordinary PC for a same algorithm. Please refer to the next Section for the current hardware performance of computing modular exponentiation and hash functions.

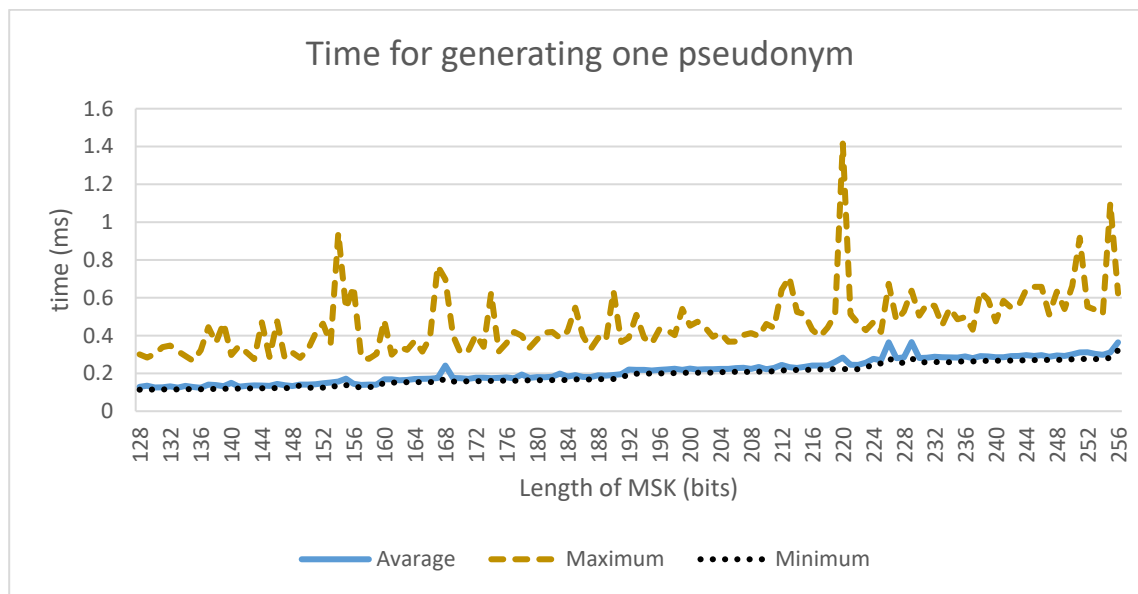


Figure 23: The performance of generating the patient's pseudonyms in computer software

7.4.2.2 Performance Estimation of Real Smart Cards

Thanks to the development of hardware technology, the computation speed of many cryptographic algorithms in smart cards is becoming faster and faster. According to the claims from some hardware design providers (Xilinx, 2015a, Kurniawan, 2002, Blum and Paar, 1999), the computation capability of modular exponentiation can easily achieve up to several hundred times per second with the exponentiation size of 256 bits, and the computation of hash functions (e.g. SHA-256) can reach 100Mbps (Xilinx, 2015b).

In our pseudonym solution and applications, each pseudonym generation needs two modular exponentiations and three hash functions (hashing on about two 256 bit-length values) according to Algorithm 2. So a smart card with such hardware can easily generate several hundred pseudonyms per second. That should be enough even if a patient has thousands of EHR entries. The patient only needs to wait no more than ten seconds to produce all of his pseudonyms if all EHR entries need to be downloaded from the cloud. The downloading of all EHR entries from the clouds should need more time according to the common speed of current network connection. Actually, the generating of pseudonyms in the smart card and the downloading of EHR entries over the network can be done simultaneously.

In our optional pseudonym solution as introduced in Algorithm 3, Algorithm 5 and Algorithm 7, only the first pseudonym generation needs computation of two modular exponentiations and three hash functions, and other pseudonyms only need computation of hash functions. The speed of generating and reproducing the pseudonyms is expected to be much faster.

7.4.3 Decreasing the Computation Load of Smart Card

To further decreasing the computation load of the smart cards of patients, we provide a solution as follows. According to Algorithm 2 and Algorithm 4 shown in chapter 5, one single pseudonym generation or reproduction needs computation of three hash computations and two modular exponentiation, where hash computation is usually fast but the modular exponentiation computation is more time consuming. Because the last two hash computations and the first modular exponentiation computation (i.e., the left top formulas in line 4 and 5 in Figure 24) do not involve the secret x , they can be computed by outside connected devices (e.g., the doctor's or the patient's computer). In a typical implementation (line 6 to 13 in Figure 24) of the modular exponentiation computation (Black et al., 2011), about two thirds of the modular multiplication computations (line 12 in Figure 24) in the second modular exponentiation computation (the blue formula in line 4 of Figure 24) will not involve the secret x , so these modular multiplication computations can also be moved to outside devices as shown in Figure 24.

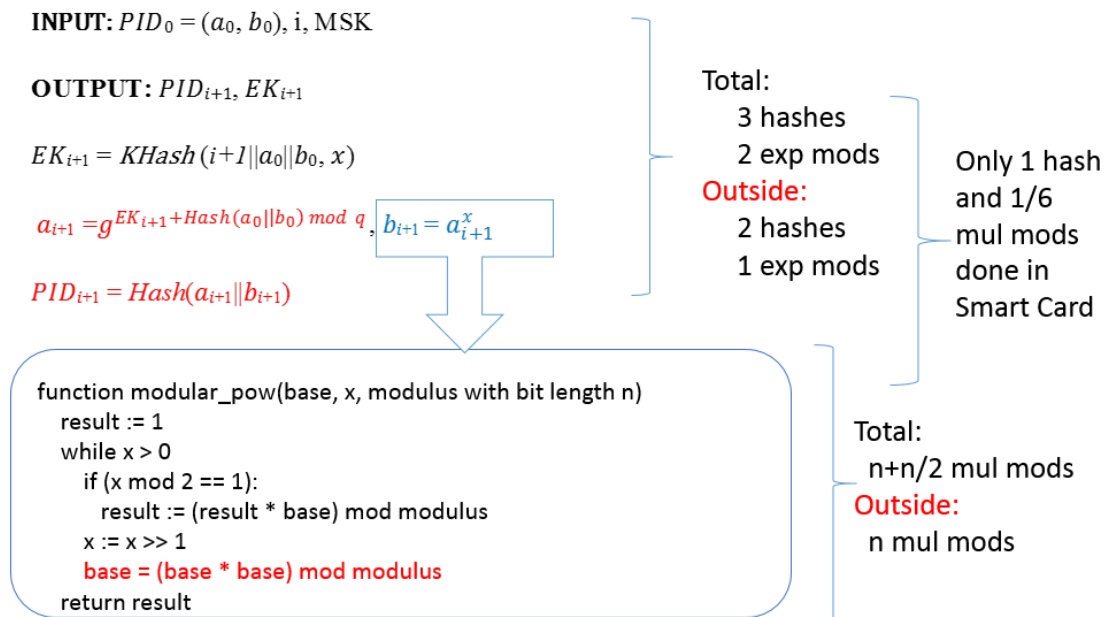


Figure 24: Use of outside devices to decrease the computation load of smart card in pseudonym generation Algorithm 2

According to the statistic in right part of Figure 24, the computation needed to be done by the smart cards is only one hash computation and a part of one modular exponentiation computation (about 1/6 modular multiplications of total). Thus, the solution can greatly decrease the computation burden of the smart cards, with the price of a bit increase of the serializable IO manipulations between smart card and the outside connected device.

7.5 Future Work for the Implementation

Supporting more activities in eHealth systems. Till now we have simulated some basic healthcare activities in our implementation. There are still more activities which are also related with the security of EHR and the privacy of patients. It is valuable to examine whether these activities can be supported by our pseudonym solution or not. Moreover, it is also important to know how these activities can be compatible with existing functions in the implementation.

Using real smart cards. The performance of the clients is very critical to the experience of a cloud-based eHealth system. It is valuable to test real smart cards in our implementation to examine the resource cost and the performance in practical healthcare activities. The actual resource cost for the algorithms proposed in this thesis determines the price of smart cards.

8 Conclusions

This thesis proposed a novel pseudonym solution which is applicable to protect the security and privacy in eHealth systems. The pseudonym solution has provable security and rule out the requirement of any powerful trusted third parties in the system. This is very helpful to dismiss the privacy worries of patients who are the real owners of the health data. Patients keep the secret keys to generate their pseudonyms, encrypt their EHR, and authorize the uses of their health data. Thus the health data are under patients' full control in every process. In the meantime, we have designed special protocols cooperating with the pseudonym solution for easing the uses of EHR in the healthcare activities, e.g. sharing EHR among doctors, secondary use.

We applied the proposed pseudonym solution for several typical healthcare activities in a cloud-based eHealth system. In these applications, the security of the health data and the privacy of patients are strongly protected by various algorithms and protocols.

Firstly, we utilized the pseudonym solution to create, store and share EHR in the ordinary healthcare activities of a cloud-based eHealth system. Each patient owns multiple pseudonyms for his/her EHR entries, and these pseudonyms are un-linkable, independent and irreversible without knowing the patient's secret key. EHR contents are encrypted by the encryption keys derived from the pseudonyms, and thus well protected. Moreover, the pseudonym solution provides an authentication algorithm. The authentication process does not disclose any identity information about patients to the verifiers. Each patient who knows the secret key can update EHR entries after being authenticated on the ownership of the pseudonyms by the cloud. With the help of the cloud, EHR can also be easily shared to any doctor who is examining the patient. We demonstrated that using the pseudonym solution in ordinary healthcare activities is sufficient in performance and profitable in security and privacy protection.

Secondly, we applied the pseudonym solution in protecting the privacy of patients from insurance companies. We refined the billing procedure in the existing eHealth systems to disclose only necessary information about the privacy of patients to insurance companies while preserving the ability to check the origin of bills. To our knowledge, patients' privacy disclosed to insurance companies has not yet attracted the attention of public. In this thesis, we at least provide a preliminary solution upon this issue.

At last, we also utilized the pseudonym solution to exploit a new model for secondary use of health data. We introduced a new consent management procedure to enable patients' full control on their own EHR when their health data are used in any other places. A communication protocol between secondary users and patients is also proposed to enable secondary users to get more information from patients and send feedback to patients. This is a new attempt to utilize the cloud for the secondary use while protecting the privacy of patients with the help of pseudonyms.

We used one unique pseudonym solution in these applications, which is beneficial to the security and privacy in eHealth systems. Each patient's secret key (MSK) is the only critical factor to protect the security and privacy in all these healthcare activities. Comparing to

those eHealth systems where there are multiple secret keys and security mechanisms to protect the security and privacy of patients, our solution decreases the risk of disclosing the secret or private information due to the weakness in a single scheme.

Since the security and privacy in eHealth systems are a challengeable topic which refers to many aspects in theory and practice, there is still much work expected to do.

Efficient structure of EHR. In this thesis, we assume that a patient generates an independent pseudonym for each EHR entry. At the end the patient will have many EHR entries with independent pseudonyms after he/she has visited doctors for many times. The independence of the pseudonyms benefits the protection of security and privacy, but it sometimes may cause trouble when a doctor wants to access some particular segments of EHR or specified type of EHR entries. Reorganizing after full downloading of all EHR entries is inefficient in the situation with slow network connection and process speed. Thus an efficient structure to organize EHR entries of each patient in the cloud is desired. Meanwhile, such kind of structure should not disclose any private information to the cloud or any other general attackers.

Fast search in the encrypted EHR. A patient or a doctor sometimes needs to search on the patient's EHR to find some specified EHR entries. It can be certainly done by searching in the decrypted EHR after downloading all the encrypted EHR entries. However, that may cause delay if the size of EHR is big and the network connection is not so fast. It is valuable to adopt a scheme for searching in the encrypted EHR entries. Then the search can be efficiently done in the encrypted data by the cloud. Some research work is being done by a researcher in our group Jan Lehnhardt (Lehnhardt et al., 2015, Lehnhardt et al., 2014). On the other side, for the sake of search in EHR by secondary users, we have suggested that EHR entries be partially encrypted as introduced in Section 6.3.3.2. The cloud can provide powerful plaintext-based search service to secondary users to filter target EHR entries. However, the partial encryption brings difficulties or fuzziness when EHR entries are created, because it might be hard to determine which parts of EHR entry are important for the search from secondary users and should be left as unencrypted without leaking the private information of patients. So it is ideal to encrypt everything when EHR entries are created in the beginning and the cloud can provide efficient search service in the encrypted data.

Further evaluation in practice. Although we have implemented a prototypical eHealth system to demonstrate the performance and feasibility of our proposed pseudonym solution, it would be better if it can be evaluated further in various aspects. At first, it is important to examine whether the performance of the solution is acceptable or not in real eHealth systems with real smart cards, because we have used some resource exhausting algorithms in our solution and the time delay is sensitive in real healthcare activities. Secondly, as we have proposed several new models or refined procedures for the use of EHR, it is uncertain that the existing business model of eHealth systems could adopt them without big earthquakes. At last, there will be a lot of unexpected problems when a new technology is deployed in practice, and we need to find out these issues to fix and enhance the existing solution.

Extension of the pseudonym solution to other information systems. Although in this thesis we mainly focused on the security and privacy issues in eHealth systems, we

noticed that there are many other potential systems which may potentially adopt our pseudonym solution. For example, it might be probable to use our pseudonym solution in the online shopping applications. The shopping information of the customers can also be stored with pseudonyms and may be secondarily used by the big-data analyzers. Another potential application scenario is the car-to-car communication. When the cars on road need to form a self-organized network, it is important to protect the real identities while sending and receiving messages. The pseudonym solution in this thesis may be applicable to deal with the privacy issues.

Bibliography

- Adrian, D., Bhargavan, K., Durumeric, Z., Gaudry, P., Green, M., Halderman, J. A., Heninger, N., Springall, D., Thomé, E. & Valenta, L. (2015). Imperfect forward secrecy: How Diffie-Hellman fails in practice. Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM, 5-17.
- Barua, M., Liang, X., Lu, R. & Shen, X. (2011). ESPAC: Enabling Security and Patient-centric Access Control for eHealth in cloud computing. International Journal of Security and Networks, 6, 67-76.
- Benzschawel, S. & Da Silveira, M. (2011). Protecting Patient Privacy when Sharing Medical Data. Proceedings of the 3rd International Conference on eHealth, Telemedicine, and Social Medicine (eTelemed), France.
- Bjurstrøm, R. & Singh, J. (2013). De-identification of Norwegian Health Record Notes. available at <http://www.diva-portal.org/smash/get/diva2:658755/FULLTEXT01.pdf>.
- Black, A. D., Car, J., Pagliari, C., Anandan, C., Cresswell, K., Bokun, T., McKinstry, B., Procter, R., Majeed, A. & Sheikh, A. (2011). The impact of eHealth on the quality and safety of health care: a systematic overview. PLoS Med, 8(1), e1000387.
- Blum, M. & Micali, S. (1984). How to generate cryptographically strong sequences of pseudorandom bits. SIAM journal on Computing, 13, 850-864.
- Blum, T. & Paar, C. (1999). Montgomery modular exponentiation on reconfigurable hardware. Computer Arithmetic. Proceedings of 14th IEEE Symposium, IEEE, 70-77.
- Borycki, E., Newsham, D. & Bates, D. (2012). eHealth in North America. Yearbook of medical informatics, 8, 103-106.
- Bouamrane, M.-M. & Mair, F. (2011). An overview of electronic health systems development & integration in Scotland. Proceedings of the first international workshop on Managing interoperability and complexity in health systems. ACM, 59-62.
- Cachin, C., Micali, S. & Stadler, M. (1999). Computationally Private Information Retrieval with Polylogarithmic Communication. International Conference on the Theory and Applications of Cryptographic Techniques. Springer Berlin Heidelberg, 402-414.
- Calisher, H. (1998). Portrait of a Pseudonym. The American Scholar, 53-61.
- Chaum, D. (1995). Designated confirmer signatures. Advances in Cryptology—EUROCRYPT'94. Springer, 86-91.
- Chaum, D. & Pedersen, T. P. (1993). Wallet databases with observers. Advances in Cryptology—CRYPTO'92. Springer, 89-105.

- Chen, L. (1996). Access with pseudonyms. *Cryptography: Policy and Algorithms*. Springer, 232-243.
- Cresswell, K. & Sheikh, A. (2009). The NHS Care Record Service (NHS CRS): recommendations from the literature on successful implementation and adoption. *Informatics in primary care*, 17, 153-160.
- Daemen, J. & Rijmen, V. (2000). The block cipher Rijndael. *Smart Card Research and Applications*. Springer, 277-284.
- Daemen, J. & Rijmen, V. (2013). *The design of Rijndael: AES-the advanced encryption standard*, Springer Science & Business Media.
- Damgård, I. B. (1990). Payment systems and credential mechanisms with provable security against abuse by individuals. *Proceedings on Advances in cryptology*. Springer-Verlag New York, Inc., 328-335.
- Deo, V., Seidensticker, R. B. & Sim, D. R. (1998). Authentication system and method for smart card transactions. Google Patents.
- Dimitrov, V. S., Jullien, G. A. & Miller, W. C. (1998). An algorithm for modular exponentiation. *Information Processing Letters*, 66, 155-159.
- Drees, D. (2007). The Introduction of Health Telematics in Germany. *ISSE/SECURE 2007 Securing Electronic Business Processes*. Springer, 396-400.
- Dubovitskaya, A., Urovi, V., Vasirani, M., Aberer, K. & Schumacher, M. I. (2015). A Cloud-Based eHealth Architecture for Privacy Preserving Data Integration. *ICT Systems Security and Privacy Protection*. Springer, 585-598.
- Eijpe, L. H. M. (2013). Overview of the national laws on electronic health records in the EU Member States. National Report for the Netherlands, available at http://ec.europa.eu/health/ehealth/docs/laws_netherlands_en.pdf.
- Elger, B. S., Iavindrasana, J., Iacono, L. L., Müller, H., Roduit, N., Summers, P. & Wright, J. (2010). Strategies for health data exchange for secondary, cross-institutional clinical research. *Computer methods and programs in biomedicine*, 99, 230-251.
- Eysenbach, G. (2001). What is e-health? *Journal of medical Internet research*, 3.
- Fan, L., Buchanan, W., Thuemmler, C., Lo, O., Khedim, A., Uthmani, O., Laws, A. & Bell, D. (2011). DACAR platform for eHealth services cloud. *Cloud Computing (CLOUD)*, 2011 IEEE International Conference. IEEE, 219-226.
- Fitzpatrick, G. (2000). Understanding the paper health record in practice: Implications for EHRs. *Proceedings of Health Informatics Conference (HIC'2000)*.
- Flajolet, P., Gardy, D. & Thimonier, L. (1992). Birthday paradox, coupon collectors, caching algorithms and self-organizing search. *Discrete Applied Mathematics*, 39, 207-229.

- Foster, I., Zhao, Y., Raicu, I. & Lu, S. (2008). Cloud computing and grid computing 360-degree compared. 2008 Grid Computing Environments Workshop, IEEE, 1-10.
- Garets, D. & Davis, M. (2006). Electronic medical records vs. electronic health records: yes, there is a difference. Policy white paper. Chicago, HIMSS Analytics, 1-14.
- Grand, J. (2004). Practical secure hardware design for embedded systems. Proceedings of the 2004 embedded systems conference, (Vol. 23).
- Grant, K. (2012). What can Canada learn from New Zealand, Denmark and the United Kingdom about Electronic Health Record (EHR) adoption. available at <https://aboutkevingrant.com/content/images/stories/sample-documents/research/analysis-paper/KevinGrant-Canadian-EHR-Learnings-From-UK-Denmark-New-Zealand.pdf>.
- Hillestad, R., Bigelow, J., Bower, A., Girosi, F., Meili, R., Scoville, R. & Taylor, R. (2005). Can electronic medical record systems transform health care? Potential health benefits, savings, and costs. *Health Affairs*, 24, 1103-1117.
- Housley, R., Polk, W., Ford, W. & Solo, D. (2002). Internet X. 509 public key infrastructure certificate and certificate revocation list (CRL) profile.
- Hsiao, C.-J., Hing, E. & Ashman, J. (2014). Trends in electronic health record system use among office-based physicians: United States, 2007-2012. *Natl Health Stat Rep*, 75, 1-18.
- Hu, Y., Lu, F., Khan, I. & Bai, G. (2012). A cloud computing solution for sharing healthcare information. *ICITST*, 465-470.
- Iacono, L. L. (2007). Multi-centric universal pseudonymisation for secondary use of the EHR. *Studies in health technology and informatics*, 126, 239.
- Jahani, S., Samsudin, A. & Subramanian, K. G. (2014). Efficient Big Integer Multiplication and Squaring Algorithms for Cryptographic Applications. *Journal of Applied Mathematics*, 2014(9).
- Jesilow, P. D., Pontell, H. N. & Geis, G. (1985). Medical criminals: Physicians and white-collar offenses. *Justice Quarterly*, 2, 149-165.
- Jha, A. K., Doolan, D., Grandt, D., Scott, T. & Bates, D. W. (2008). The use of health information technology in seven nations. *International journal of medical informatics*, 77, 848-854.
- Jin, J., Ahn, G.-J., Hu, H., Covingt, M. J. & Zhang, X. (2009). Patient-centric authorization framework for sharing electronic health records. Proceedings of the 14th ACM symposium on Access control models and technologies, ACM, 125-134.
- Jones, S. S., Adams, J. L., Schneider, E. C., Ringel, J. S. & McGlynn, E. A. (2010). Electronic health record adoption and quality improvement in US hospitals. *The American journal of managed care*, 16, 64-71.

- Katz, J. & Lindell, Y. (2014). Introduction to modern cryptography, CRC Press.
- Knuth, D. E. (1981). Seminumerical Algorithms, The Art of Computer Programming (2nd ed.), Addison-Wesley, Reading, MA.
- Krawczyk, H., Canetti, R. & Bellare, M. (1997). HMAC: Keyed-hashing for message authentication. Available: <https://tools.ietf.org/html/rfc2104>
- Kurniawan, B. (2002). ASIC design and implementation of a parallel exponentiation algorithm using optimized scalable Montgomery multipliers. Available: <http://hdl.handle.net/1957/32189>
- Lassus, M. (1997). Smart-cards-a cost-effective solution against electronic fraud. Security and Detecti, 1997. ECOS 97., European Conference, IET, 81-85.
- Lehnhardt, J., Rho, T., Spalka, A. & Cremers, A. B. (2014). Ordered Range Searches on Encrypted Data. In: Technical Report IAI-TR-2014-03, Computer Science Department III, University of Bonn, ISSN 0944-8535.
- Lehnhardt, J., Rho, T., Spalka, A. & Cremers, A. B. (2015). Performance-optimized Indexes for Inequality Searches on Encrypted Data in Practice. 1st International Conference on Information Systems Security and Privacy (ICISSP 2015), 221-229.
- Lehnhardt, J. & Spalka, A. (2011). Decentralized Generation of Multiple, Uncorrelatable Pseudonyms without Trusted Third Parties. In: Trust, Privacy and Security in Digital Business. Springer Berlin Heidelberg, 113-124.
- Li, Z.-R., Chang, E.-C., Huang, K.-H. & Lai, F. (2011). A secure electronic medical record sharing mechanism in the cloud computing platform. Consumer Electronics (ISCE), 2011 IEEE 15th International Symposium, IEEE, 98-103.
- Lim, C. H. & Lee, P. J. (1997). A key recovery attack on discrete log-based schemes using a prime order subgroup. Advances in Cryptology—CRYPTO'97. Springer, 249-263.
- Lu, R., Li, X., Luan, T. H., Liang, X. & Shen, X. (2012). Pseudonym changing at social spots: An effective strategy for location privacy in vanets. Vehicular Technology, IEEE Transactions, 61, 86-96.
- Lysyanskaya, A., Rivest, R. L., Sahai, A. & Wolf, S. (2000). Pseudonym systems. Selected Areas in Cryptography, Springer, 184-199.
- Martin-Löf, P. (1966). The definition of random sequences. Information and control, 9, 602-619.
- Mazieres, D. & Kaashoek, M. F. (1998). The design, implementation and operation of an email pseudonym server. Proceedings of the 5th ACM Conference on Computer and Communications Security, ACM, 27-36.
- McCurley, K. S. (1990). The discrete logarithm problem. Proc. of Symp. in Applied Math, 49-74.

- Mell, P. & Grance, T. (2011). The NIST definition of cloud computing.
- Menezes, A. J., Van Oorschot, P. C. & Vanstone, S. A. (1996). Handbook of applied cryptography, CRC press.
- Milroy, M. & Li, F. (2001). Internet billing: the experience from four UK utility companies. *International journal of information management*, 21, 101-121.
- Mollin, R. A. (2006). An introduction to cryptography, CRC Press.
- Molnar, D., Soppera, A. & Wagner, D. (2006). A scalable, delegatable pseudonym protocol enabling ownership transfer of RFID tags. *Selected Areas in Cryptography*, Springer, 276-290.
- Nagy, M., Preckova, P., Seidl, L. & Zvarova, J. (2010). Challenges of interoperability using hl7 v3 in czech healthcare. *Studies in health technology and informatics*, 155, 122-128.
- Narayan, S., Gagné, M. & Safavi-Naini, R. (2010). Privacy preserving EHR system using attribute-based infrastructure. *Proceedings of the 2010 ACM workshop on Cloud computing security workshop*, ACM, 47-52.
- Needham, R. M. & Schroeder, M. D. (1978). Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21, 993-999.
- Neubauer, T. & Heurix, J. (2011). A methodology for the pseudonymization of medical data. *International journal of medical informatics*, 80, 190-204.
- NIST (2002). Secure Hash Standard, FIPS PUB 180-2.
- Noumeir, R., Lemay, A. & Lina, J.-M. (2007). Pseudonymization of radiology data for research purposes. *Journal of digital imaging*, 20, 284-295.
- Oh, H., Rizo, C., Enkin, M. & Jadad, A. (2005). What is eHealth (3): a systematic review of published definitions. *Journal of medical Internet research*, 7.
- Pfitzmann, A. & Köhntopp, M. (2001). Anonymity, unobservability, and pseudonymity—a proposal for terminology. *Designing privacy enhancing technologies*, Springer, 1-9.
- Pommerening, K. & Reng, M. (2004). Secondary use of the EHR via pseudonymisation. *Studies in health technology and informatics*, 441-446.
- Protti, D. (2007). US regional health information organizations and the nationwide health information network: any lessons for Canadians? *Healthcare quarterly (Toronto, Ont.)*, 11, 96-101, 4.
- Quinn, J. (1998). An HL7 (Health Level Seven) overview. *Journal of AHIMA/American Health Information Management Associati*, 70, 32-4; quiz 35-36.
- Rankl, W. & Effing, W. (2010). Smart card handbook, John Wiley & Sons.

- Rau, H.-H., Hsu, C.-Y., Lee, Y.-L., Chen, W. & Jian, W.-S. (2010). Developing electronic health records in Taiwan. *IT professional*, 17-25.
- Ravi Chandra, J. & Sharad, M. (2006). Querying Encrypted XML Documents. *Database Engineering and Applications Symposium. IDEAS '06. 10th International*, Dec. 2006. 129-136.
- Reed, M. G., Syvers, P. F. & Goldschlag, D. M. (1996). Proxies for anonymous routing. *Computer Security Applications Conference, 1996., 12th Annual, IEEE*, 95-104.
- Riedl, B., Grascher, V., Fenz, S. & Neubauer, T. (2008). Pseudonymization for improving the privacy in e-health applications. *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual, IEEE*, 255-255.
- Rogaway, P. & Shrimpt, T. (2004). Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. *Fast Software Encrypti, Springer*, 371-388.
- Samarati, P. & Sweeney, L. (1998). Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, SRI International.
- Schabetsberger, T., Ammenwerth, E., Andreatta, S., Gratl, G., Haux, R., Lechleitner, G., Schindelwig, K., Stark, C., Vogl, R. & Wilhelmy, I. (2006). From a paper-based transmission of discharge summaries to electronic communication in health care regions. *International journal of medical informatics*, 75, 209-215.
- Schneier, B. (1999). Attack trees. *Dr. Dobb's journal*, 24, 21-29.
- Seo, D. H. & Sweeney, P. (1999). Simple authenticated key agreement algorithm. *Electronics Letters*, 35, 1073-1074.
- Shamir, A. (1985). Identity-based cryptosystems and signature schemes. *Proceedings of CRYPTO 84 on Advances in cryptology. Santa Barbara, California, United States: Springer-Verlag New York, Inc.*, 47-53.
- Shekelle, P., Mort, S. C. & Keeler, E. B. (2006). Costs and benefits of health information technology. Available: <https://www.ncbi.nlm.nih.gov/books/NBK37988/>.
- Shokri, R., Theodorakopoulos, G., Le Boudec, J.-Y. & Hubaux, J.-P. (2011). Quantifying location privacy. *Security and Privacy (SP), 2011 IEEE Symposium, IEEE*, 247-262.
- Shoup, V. (1997). Lower bounds for discrete logarithms and related problems. *International Conference on the Theory and Applications of Cryptographic Techniques, Springer*, 256-266.
- Shu, T., Liu, H., Goss, F. R., Yang, W., Zhou, L., Bates, D. W. & Liang, M. (2014). EHR adoption across China's tertiary hospitals: a cross-sectional observational study. *Int J Med Inform*, 83, 113-21.

- Slamanig, D. & Stingl, C. (2008). Privacy aspects of ehealth. Availability, Reliability and Security, 2008. ARES 08. Third International Conference, IEEE, 1226-1233.
- Stroetmann, K. A., Artmann, J., Stroetmann, V. N. & Whitehouse, D. (2011). European countries on their journey towards national eHealth infrastructures. European commission DG information society and media ICT for health unit.
- Sunyaev, A., Chorny, D., Mauro, C. & Krcmar, H. (2010). Evaluation framework for personal health records: Microsoft HealthVault vs. Google Health. System Sciences (HICSS), 2010 43rd Hawaii International Conference, IEEE, 1-10.
- Tang, P. C., Ash, J. S., Bates, D. W., Overhage, J. M. & Sands, D. Z. (2006). Personal health records: definitions, benefits, and strategies for overcoming barriers to adoption. Journal of the American Medical Informatics Association, 13, 121-126.
- Tanner, A. (2014). What Stays in Vegas: The World of Personal Data - Lifeblood of Big Business - and the End of Privacy as We Know It, PublicAffairs, ISBN13: 9781610394185.
- Tuecke, S., Welch, V., Engert, D., Pearlman, L. & Thomps, M. (2004). Internet X. 509 public key infrastructure (PKI) proxy certificate profile.
- Van Dijk, L. V., De Vries, H. & Bell, D. S. (2011). Electronic Prescribing in the United Kingdom and in the Netherlands. Prepared for: Agency for Healthcare Research and Quality US Department of Health and Human Services, 540.
- Vaquero, L. M., Rodero-Merino, L., Caceres, J. & Lindner, M. (2008). A break in the clouds: towards a cloud definition. ACM SIGCOMM Computer Communication Review, 39, 50-55.
- Viega, J., Messier, M. & Chandra, P. (2002). Network Security with OpenSSL: Cryptography for Secure Communications, " O'Reilly Media, Inc.".
- Waegemann, C. P. (2003). Ehr vs. cpr vs. emr. Healthcare Informatics Online, 1-4.
- Wang, C. & Wulf, W. A. (1997). Towards a framework for security measurement. 20th National Information Systems Security Conference, Baltimore, MD, 522-533.
- Whitehouse, D., Giest, S., Domortier, J., Artmann, J. & Heywood, J. (2010). Contry Brief: England. eHealth Strategies, http://ehealth-strategies.eu/database/documents/England_CountryBrief_eHStrategies.pdf.
- Wilken, T. (2014). PaaS Reference Implementation of a pseudonymity based eHealth cloud system. Diploma Thesis of University of Bonn.
- Xilinx. (2015). Modular Exponentiation Engine for RSA and DH (ModExp). Available: <http://www.xilinx.com/products/intellectual-property/1-8dyf-1344.html>.
- Xilinx. (2015). Tiny Hashing Cores. Available: http://www.heliontech.com/tiny_hash.htm.

- Xu, L. & Cremers, A. B. (2014). A Decentralized Pseudonym Scheme for Cloud-based eHealth Systems. Proc. International Conference on Health Informatics, Angers, France, 230-237.
- Xu, L. & Cremers, A. B. (2014). Patients' Privacy Protection against Insurance Companies in eHealth Systems. Secure IT Systems, Springer, 247-260.
- Xu, L., Cremers, A. B. & Wilken, T. (2014). Pseudonymization for Secondary Use of Cloud Based Electronic Health Records. 2014 ASE BigData/SocialInformatics/PASSAT /BioMedCom Conference. Harvard University.
- Yang, T.-H., Cheng, P.-H., Yang, C., Lai, F., Chen, C., Lee, H.-H., Hsu, K.-P., Chen, C.-H., Tan, C.-T. & Sun, Y. S. (2006). A scalable multi-tier architecture for the National Taiwan University hospital information system based on HL7 standard. Computer-Based Medical Systems, 2006. CBMS 2006. 19th IEEE International Symposium, IEEE, 99-104.
- Yao, A. C.-C. (1982). Protocols for secure computations. Foundations of Computer Science, SFCS'08. 23rd Annual Symposium, 160-164.
- Yo, J. W. & Kim, H. (2011). A perfect collision-free pseudonym system. Communications Letters, IEEE, 15, 686-688.
- Zhang, R. & Liu, L. (2010). Security models and requirements for healthcare application clouds. Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference, IEEE, 268-275.

List of Figures

Figure 1: A general model of eHealth systems.....	20
Figure 2: Patient registers at an insurance company	24
Figure 3: A patient visits a doctor in hospital	26
Figure 4: Two typical business models of billing	27
Figure 5: Process of secondary use of EHR	29
Figure 6: Two models of storing EHR.....	30
Figure 7: Examples for general attackers to EHR	34
Figure 8: Privacy disclosure when a secondary user sends feedback to a patient	38
Figure 9: An eHealth model for ordinary healthcare activities and secondary use.....	40
Figure 10: Using pseudonyms in the ordinary healthcare activities	61
Figure 11: A Simple model of billing procedure with smart card	71
Figure 12: A cloud-baded eHealth system enabling ordinary healthcare activities and secondary use.....	80
Figure 13: Architecture of the prototypical eHealth system	90
Figure 14: A patient sets up the PIN and secret key	91
Figure 15: A doctor creates an EHR entry for the patient.....	92
Figure 16: EHR of the patient is viewed	92
Figure 17: A pharmacy sells the medicines in the prescription to the patient.....	93
Figure 18: A secondary user views all EHR entries from the cloud.....	94
Figure 19: A secondary user sends the requests to the patient	95
Figure 20: The patient replies to the secondary user	96
Figure 21: The secondary user views the responses from the patient	97
Figure 22: The performance in initializing a patient's MSK with computer software	99
Figure 23: The performance of generating the patient's pseudonyms in computer software	100
Figure 24: Use of outside devices to decrease the computation load of smart card in pseudonym generation Algorithm 2.....	101
Figure 25: Load tests on the cloud throughput (Wilken, 2014).....	117
Figure 26: Load tests on the database (Wilken, 2014).....	118

List of Tables

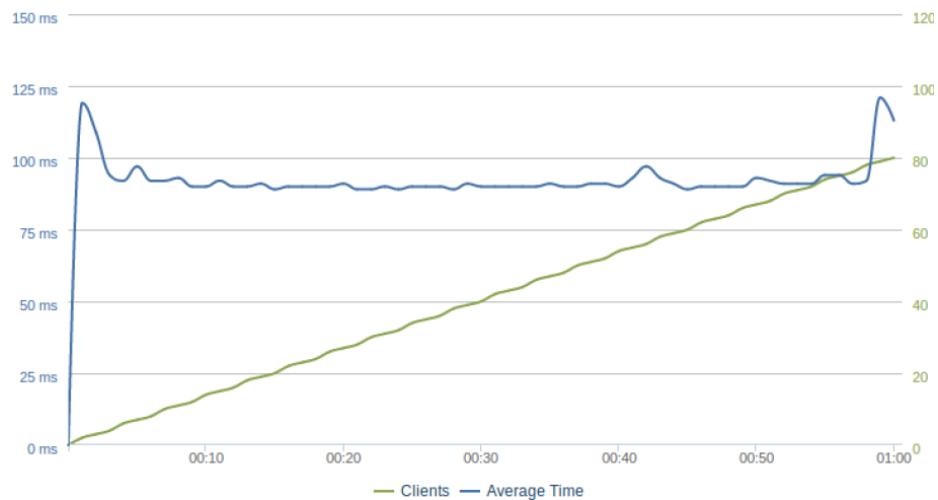
Table 1: An example of health records with real identities..... 7
Table 2: An example of anonymizing the health records 8
Table 3: An example of pseudonymizing the health records 8
Table 4: An example to re-identify the patient in secondary use 37
Table 5: A comparison of existing pseudonym solutions in eHealth systems..... 49

List of Algorithms

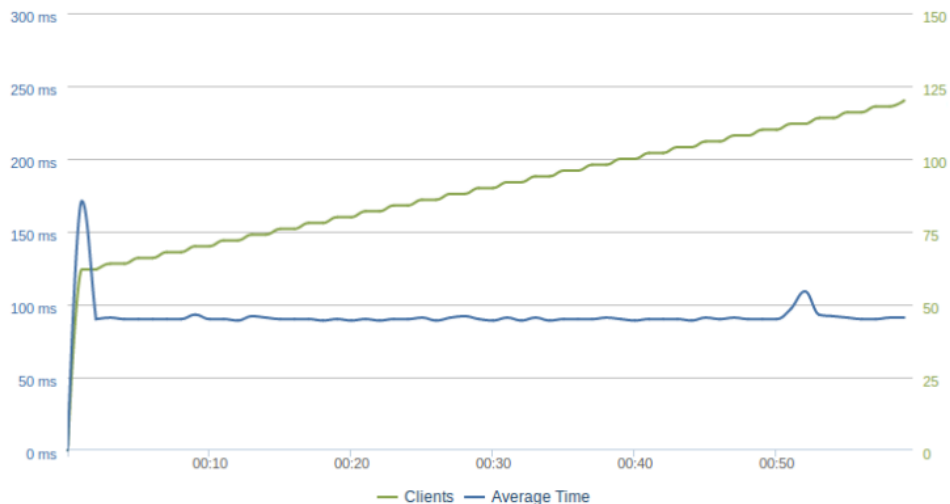
Algorithm 1: Generating the user’s MSK..... 53
Algorithm 2: Generating the user’s pseudonym..... 54
Algorithm 3: An optional algorithm for generating a user’s pseudonyms 55
Algorithm 4: Reproducing the user’s pseudonym 56
Algorithm 5: An Optional way of reproducing a user’s pseudonyms..... 56
Algorithm 6: Algorithm for authentication of PID 57
Algorithm 7: An optional algorithm for authentication of PID 57
Algorithm 8: Algorithm for authenticating with less messages 58
Algorithm 9: Generating a bill..... 73
Algorithm 10: Patient signs the bill with pseudonym 75
Algorithm 11: Insurance company verify the pseudonymous signature on the bill 76

Appendix: Load Tests on Cloud and Database

To examine how many simultaneous requests can be handled by the cloud where our applications reside in, different load tests are conducted for different application scenarios. These load tests provide the application throughput and give suggestions for the scaling up or down the resources depending on the load. In Figure 25, we give two examples of our tests for simultaneous access to the cloud by the clients. We define a successful response as returning result without any errors. The results of the two tests show that the average response time of cloud stays almost stable with the proper scaling of resources.



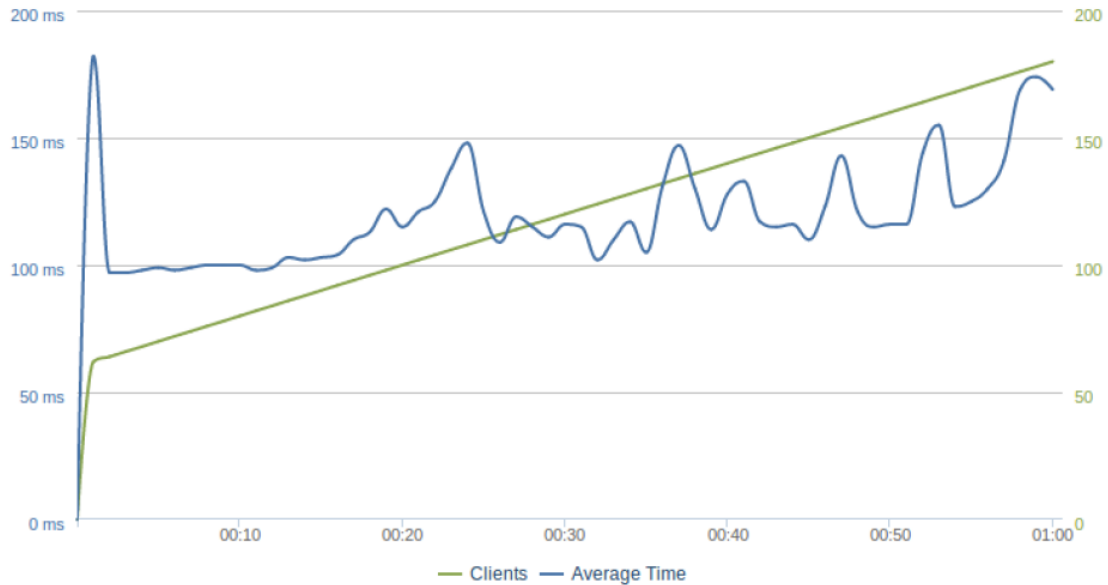
- (a) One minute load test with 10 to 80 clients on a three 512 MB container setup. Within the last seconds the average time increases, giving a hint the the system reaches its resource limit.



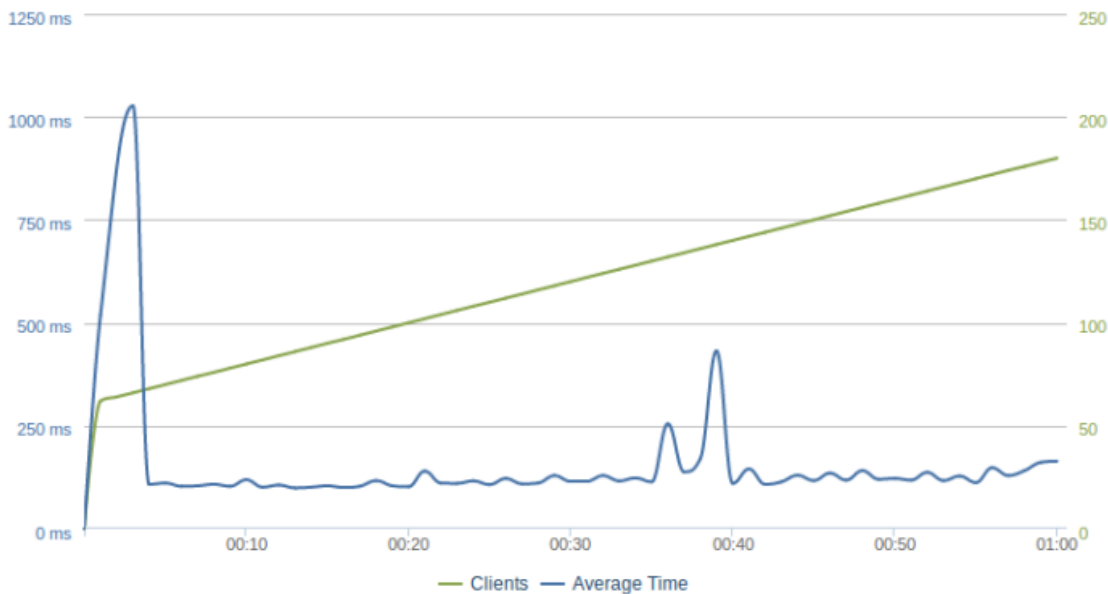
- (b) One minute load test with 60 to 120 clients on a six 512 MB container setup, with a stable response time.

Figure 25: Load tests on the cloud throughput (Wilken, 2014)

To examine the performance of MongoDB residing in the cloud, we also simulate a scenario where a lot of users are creating and storing EHR entries to the database simultaneously. From the test results in Figure 26, we can find out that the average response time of the database stays almost stable with the proper scaling of resources.



(a) One minute load test for a single MongoDB server setup from 60 to 180 clients. Up to 80 clients the response time stays stable. More clients result in fluctuating results with up to 173 ms response time.



(b) One minute load test for a three server sharded MongoDB setup from 60 to 80 clients. The response time stays mostly stable. The peak around 00:38 can be explained by a new allocation of data files on one of the servers.

Figure 26: Load tests on the database (Wilken, 2014)