Secure and Privacy-Preserved Solutions for Distributed Electronic Health Systems

Fatemeh Rezaeibagha
University of Wollongong
Secure and Privacy-Preserved Solutions for Distributed Electronic Health Systems

A thesis submitted in fulfillment of the requirements for the award of the degree

Doctor of Philosophy

from

UNIVERSITY OF WOLLONGONG

by

Fatemeh Rezaeibagha

School of Computing and Information Technology
November 2017
Dedicated to

My family
Declaration

This is to certify that the work reported in this thesis was done by the author, unless specified otherwise, and that no part of it has been submitted in a thesis to any other university or similar institution.

Fatemeh Rezaeibagha
November 6, 2017
Abstract

With the development of online services, the traditional paper-based healthcare services are replaced by the Electronic Health Record System (EHRS) that has contributed significantly to the improvement of individual well-being and public health. In recent years, advances in EHRS have ameliorated the integration among various medical practitioners and healthcare givers where medical data could be accessed more conveniently. This has not only accelerated decision-making procedures but also saved the users time and money vastly. However, the adoption of EHRS has arisen a common concern about security and privacy as EHR accumulates sensitive health data. Therefore, protection of patient privacy and security of EHR must be considered in designing the EHRS. Although a number of mature cryptographic tools could be adopted, the nature of complexity of EHRS and sophistical data access requirements among medical stakeholders in EHRS have made the tasks challenging.

While one of the advantages of EHRS is data sharing, it poses difficulties about how to control data sharing so that security and privacy can be ensured. In this thesis, we present several novel techniques, which can help to solve some critical problems we have identified in EHRS. One of the major tools we developed in this thesis is novel access control technologies for EHRS to solve the security and privacy issues. The challenge we face is that EHRS is usually operated in a distributed environment. Although we need to ensure flexibility and scalability in data sharing, data security against potential attacks must be achieved. Traditional access control systems are not sufficient.

In this thesis, we adopt novel encryption techniques such as attribute-based encryption and authenticated encryption to achieve access control for the special needs of EHRS. We allow multiple authorities to better manage the distributed EHRS such as those operated in the cloud. We present the security protocols in order to demonstrate how to apply our approaches to real world EHR application.

As an important part of access control technology, access control policies are the
core of the entire system. We investigate various access control policies for EHRS. We present a policy integration approach as a novel solution based on a policy similarity, which has provided a new way for EHRS in cloud computing, where two or more access control policies can be integrated in order to suit the need of policy management. We use XACML as an example to show how this can be done in practice. We also provide a novel approach for access control policy transformation in cloud computing, where the policy for the private patient records in a private cloud can be transformed into a different policy which can handle access rights for different stakeholders.

This thesis also covers the user mobility issues in EHRS. We proposed several security protocols that capture secure communication between patients and doctors who are located in different locations. Our proposed protocols achieve authentication, confidentiality and anonymity features in remote telemedicine systems. Our protocols are the first of this kind, which provide sound solutions to user mobility in EHRS.

Within the scope of this thesis, we present an approach to manage a patient monitoring system in order to provide efficient authentication and confidentiality to patient data transmission. Again, we assume that our system is set up in a distributed environment. We propose a new signcryption scheme which offers the feature of homomorphism. Therefore, the signencrypted patient data items can be automatically aggregated without the need of decryption. Our scheme is the first provably secure homomorphic signcryption scheme, in that the previous solution is not provably secure.
Acknowledgement

I would like to express my deepest thanks to my principal supervisor, Professor Yi Mu, for his support, understanding, immense knowledge and invaluable guidance to my research. He has given me invaluable suggestions and encouragement from the beginning of my research career. This thesis would have been impossible to accomplish without his support. I can not find words grand enough to say Thank You for all of your support and understanding. In addition, I would like to thank my co-supervisor, Dr Fuchun Guo, for his support in my research.

I wish to thank all my lovely and fellow students during my PhD study at UOW. The non-exhaustive list of whom includes: Dr Shiwei Zhang, Dr Yangguang Tian, Dr Clementine Gritti, Dr Man Ho (Allen) Au, Dr Yong Yu, Prof. Xiaofen Wang, Dr Rongmao Chen, Sao Ian Leong, Dr Hui Cui, Yinhao Jiang, Shengmin Xu, Dr Weiwei Liu, Dr Jiannan Wei, Jianchang Lai, Zhongyuan Yao, Ge Wu, Tong Wu, Dr. Nan Li, Dr Goumin Yang and Xueqiao Liu.

It is my honour to be a PhD student of the institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong.

Finally, I am sincerely grateful to my family for their love, support and encouragement: my mother, Mehrangiz, who has persuaded me to enrol in PhD study, my sister, Marzieh, who has supported me emotionally and heartily, and my father, Hassan, for everything. I should recall my grandpa, who was the main supporter of my study abroad. If he could see my PhD testimony, he would be so happy and proud of his grand daughter. All my achievements in my study would be impossible without their unconditional support.

Fatemeh Rezaeibagha

August, 2017
Wollongong
Publications

This thesis is based on the following presented or published papers, which were finished when I was in pursuit of the PhD degree in University of Wollongong.


Papers which are under review:

5. Fatemeh Rezaeibagha, Yi Mu and Shiwei Zhang, Provably secure homomorphic sign-encryption and its application, Provsec 2017, Lecture Notes in Computer Science, Springer


viii
Contents

Abstract v
Acknowledgement vii
Publications viii
List of Abbreviations 1

1 Introduction 2
  1.1 Security and Privacy in EHRS . . . . . . . . . . . . . . . . . . . . . . 2
  1.2 Motivation of research . . . . . . . . . . . . . . . . . . . . . . . . . 3
  1.3 Thesis Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . 7
  1.4 Thesis Organisation . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

2 Preliminaries 11
  2.1 Abstract Algebra . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
  2.2 Foundations of Algebra . . . . . . . . . . . . . . . . . . . . . . . . . 11
    2.2.1 Field . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
    2.2.2 Group . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
  2.3 Bilinear Groups . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
  2.4 Complexity Assumptions . . . . . . . . . . . . . . . . . . . . . . . . 14
  2.5 Cryptographic Tools . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
    2.5.1 Hash Function . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
    2.5.2 Random Oracle Model . . . . . . . . . . . . . . . . . . . . . . 16
    2.5.3 Public-Key Encryption . . . . . . . . . . . . . . . . . . . . . . 17
    2.5.4 Sequences of Games . . . . . . . . . . . . . . . . . . . . . . . 19
    2.5.5 Digital Signature . . . . . . . . . . . . . . . . . . . . . . . . . 21
    2.5.6 Signcryption . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23
2.6 An overview of XACML

3 Multi-Authority Security Framework for Scalable EHR Systems

3.1 Introduction

3.1.1 Health IT Standards

3.1.2 Related Work

3.2 Model

3.3 ATTRIBUTE-BASED SECURITY FRAMEWORK WITH MULTIPLE TRUST AUTHORITIES

3.4 THE PROTOCOLS

3.4.1 EHR System Setup

3.4.2 Registration

3.4.3 Patient Record Management

3.5 AN APPLICATION SCENARIO

3.6 CONCLUSION

4 Distributed Healthcare Information Sharing with Dynamic Access Control Policy Transformation

4.1 Introduction

4.2 Related Work

4.3 Proposed Architecture

4.3.1 Data Sharing Scenario

4.3.2 Our Proposed Architecture

4.3.3 Definitions

4.3.4 Access Control and Transformation of EHR Policy

4.4 An Instance of Application

4.5 Discussions and Remarks

4.6 Conclusion

5 Access Control Policy Combination from Similarity Analysis for Secure Privacy-preserved EHR Systems

5.1 Introduction

5.2 Related work

5.3 EHR system scenario

5.4 Proposed policy combination system

5.4.1 Definitions
5.4.2 Policy similarity ............................................. 76
5.4.3 Policy combination ........................................... 79
5.5 A Case study ..................................................... 82
5.6 Conclusion ....................................................... 85

6 Practical and Secure Telemedicine Systems for User Mobility 86
6.1 Introduction .................................................... 86
6.2 System Model and Security Requirements ....................... 88
  6.2.1 System Model .............................................. 88
  6.2.2 Security Requirements and Notations ..................... 89
  6.2.3 System Setup .............................................. 90
6.3 Protocols ....................................................... 92
  6.3.1 Scenario 1 ................................................ 92
  6.3.2 Scenario 2 ................................................ 93
  6.3.3 Scenario 3 ................................................ 96
  6.3.4 Scenario 4 ................................................ 98
6.4 Routing Tables ............................................... 100
6.5 Clinical Examples ............................................. 100
6.6 Security Analysis ............................................. 103
6.7 Conclusion ..................................................... 105

7 Provably Secure Homomorphic Sign-encryption and Its Application in Patient Monitoring 106
7.1 Introduction .................................................... 106
7.2 Related work ................................................... 109
7.3 Definitions and Models ......................................... 111
  7.3.1 System model ............................................. 111
  7.3.2 Complexity Assumptions .................................. 112
  7.3.3 The Definition of Homomorphic Sign-encryption Scheme .. 113
  7.3.4 Security model ............................................ 114
7.4 Our Proposed Scheme ........................................ 115
7.5 Security Analysis ............................................. 116
7.6 Conclusion ..................................................... 120

8 Conclusion and Future Work ................................... 121
List of Tables

3.1 Properties of the related work and our framework. . . . . . . . . . . . . 38
3.2 The format of patient records. . . . . . . . . . . . . . . . . . . . . . . . . . 44
3.3 Access arrangement. Where Patient Record Server could be TA . . . . . 45
3.4 The access table for Mike. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50
4.1 The private cloud access table. . . . . . . . . . . . . . . . . . . . . . . . . 61
4.2 The public cloud access table. . . . . . . . . . . . . . . . . . . . . . . . . . 61
5.1 Deny Override (ordered and unordered) . . . . . . . . . . . . . . . . . . . 81
5.2 Permit Override (ordered and unordered) . . . . . . . . . . . . . . . . . . 82
5.3 First-applicable . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 82
5.4 Only-One Applicable . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 83
5.5 The instances of pol and pol’ written with XACML . . . . . . . . . . . . 84
5.6 Combined policies . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 85
6.1 Comparison of the proposed scheme with related works. . . . . . . . . . 88
6.2 Notations . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 90
6.3 Example of routing table. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 100
List of Figures

2.1 XACML 3.0 Policy Language Model. 26
2.2 XACML Data Flow Diagram. 26

3.1 System model. 39
3.2 Trust Relationship Diagram. 40
3.3 Data Flow Diagram. 41

4.1 The framework with policy transformation. 58

5.1 EHR Policy combination Scenario. 73
5.2 Policy Combination Framework in XACML. 74

6.1 System model. 89
6.2 Scenario 1. The figure on the left-hand side is for the patient ticket granting. The figure on the right-hand side is for the consultation session. 92
6.3 Scenario 2. The figure on the left-hand side shows the case when the patient has travelled to the remote domain, while the doctor is still in its home domain. The figure on the right-hand side shows the consultation phase, which is done with the assistance of HS and RS. 94
6.4 Scenario 3. The figure on the left-hand side shows that the patient and the doctor both have left HS and are now located at the remote domain managed by RS. The figure on the right-hand side shows the consultation communication flows. 96
6.5 Scenario 4. The figure on the left-hand side shows the case while both P and D have a separate home server and both reside in their own home. The figure on the right-hand side shows the consultation phase. 98

7.1 System Model. 112
List of Abbreviations

The following abbreviations are used throughout this thesis. Some special abbreviations will be defined when they are used for first time.

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABAC</td>
<td>Attribute-based Access Control;</td>
</tr>
<tr>
<td>ABE</td>
<td>Attribute-based Encryption;</td>
</tr>
<tr>
<td>ABPRE</td>
<td>Attribute-based Proxy Re-Encryption;</td>
</tr>
<tr>
<td>KP-ABE</td>
<td>Key-Policy Attribute-based Encryption;</td>
</tr>
<tr>
<td>CA</td>
<td>Certificate Authority;</td>
</tr>
<tr>
<td>CP-ABE</td>
<td>Ciphertext-Policy Attribute-based Encryption;</td>
</tr>
<tr>
<td>CDH</td>
<td>Computational Diffie-Hellman;</td>
</tr>
<tr>
<td>DDH</td>
<td>Decisional Diffie-Hellman;</td>
</tr>
<tr>
<td>DLP</td>
<td>Discrete Logarithm Problem;</td>
</tr>
<tr>
<td>EMR</td>
<td>Electronic Medical Record;</td>
</tr>
<tr>
<td>EHR</td>
<td>Electronic Health Record;</td>
</tr>
<tr>
<td>EU-CMA</td>
<td>Existentially Unforgeable under Chosen-message Attacks;</td>
</tr>
<tr>
<td>IND-CPA</td>
<td>Indistinguishability against Adaptive Chose Plaintext Attacks;</td>
</tr>
<tr>
<td>PCL</td>
<td>Policy Combining Language;</td>
</tr>
<tr>
<td>PKE</td>
<td>Public Key Encryption;</td>
</tr>
<tr>
<td>RBAC</td>
<td>Role-based Access Control;</td>
</tr>
<tr>
<td>PRE</td>
<td>Proxy Re-Encryption;</td>
</tr>
<tr>
<td>SKE</td>
<td>Symmetric Key Encryption;</td>
</tr>
<tr>
<td>TA</td>
<td>Trusted Authority;</td>
</tr>
<tr>
<td>TE</td>
<td>Threshold encryption;</td>
</tr>
<tr>
<td>TMIS</td>
<td>Telecare Medical Information Systems;</td>
</tr>
<tr>
<td>XACML</td>
<td>eXtensible Access Control Markup Language.</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Security and Privacy in EHRS

Electronic health record systems (EHRS) have gained increasing attention due to the fact that they have prominent impacts on the efficiency of managing and improving the individual’s health. In these systems, traditional paper-based healthcare is replaced with comprehensive electronic records which offer more benefits to humans than traditional procedures. With the adoption of EHRS, healthcare givers are enabled to integrate patient records from different regions and leverage digital processes, in which health information can be accessed whenever and wherever they are needed. This has benefited enormously to patient wellness and public health.

An electronic health record (EHR) involves different types of information (e.g. medication, laboratory test results, medical history), and benefits doctors, patients, healthcare providers and governments in several aspects. In an EHRS, patient records can be shared across authorised stakeholders, enabling patients to receive heterogeneous medical services efficiently and effectively. Developments of distributed systems lead to improvement in the communication among users and patients’ treatment procedures on the grounds that medical records are accessible to intended doctors at any time. This not only helps doctors in better decision making but also reduces the barrier of time, cost, and location, whereas these challenges have been unavoidable in traditional medical systems. It also benefits the governments through improvement in the level of public health and economic state of the society. Although numerous benefits of EHRS have been identified, security and privacy requirements have remained challenging issues in EHR systems. It is important to build an EHRS that also achieves security and privacy to prevent the contents of EHR systems from disclosing to the unauthorised parties.
1.2 Motivation of research

There are existing impediments to the implementation of secure and privacy-preserved EHRS. For example, patient medical history must not be accessed by unauthorised parties other than the authorised doctors and the corresponding patient. While, in some circumstances, patient records are needed to be accessed to physicians, such as in emergency conditions, and might be required to be shared to offer comprehensive healthcare services in an ubiquitous environment. In large-scale healthcare systems, an EHR system is often operated in a distributed environment; therefore, multiple authorities and users with various access levels are involved. Hence, a much more sophisticated security design has to be in place.

To achieve security and privacy in EHRS, designing fine-grained access control mechanisms play a fundamental role. The implementation of access control mechanisms in the system limits the data access to only authorised users, preventing any unauthorised accesses to data. There are various types of access control models, including attribute-based access control (ABAC), role-based access control (RBAC), and cryptographic access control (CAC) [RWS15]. There are also pre-defined access control policies and standards to assure secure access to users’ data, such as ISO standards, XACML, HIPAA, etc; however, flexibility in accessing patient’s data has been accounted as one of the desired features in the security designs.

The access control designs are required to offer flexibility while the system is distributed or shared among different enterprises. Then, secure data sharing can increase the flexibility and availability of delivery of care where users are located at different places. Secure data sharing relies on cryptography, which offers various security services including authentication, data integrity and confidentiality. Security implementations are required to consider the scalability of medical systems in remote area networks or distributed environments where databases are not centralised. Notably, the data exchanging on the communication channels has to be protected against any modification and loss.

1.2 Motivation of research

Despite previous attempts to secure electronic health record systems, there are still increasing demands to improve security and privacy. Our motivation of research mainly focuses on improving security and privacy in EHRS in terms of realistic applications such as cloud computing, distributed databases, etc. In the following, we present some desirable security requirements for EHRS we consider.
Confidentiality:

Storing/retrieving unprotected EHRs on/from networked databases poses major threats to health data security due to the disclosure of crucial data to various forms of online and offline adversaries. The unprotected patient’s information stored on local or distributed databases can be easily accessed and modified by unauthorised people. Any modification of patient’s medical status or medical history leads to fatal consequences. In order to prevent unauthorised accesses and ensure confidentiality, encryption techniques need to be employed. Cryptography plays an important role in ensuring the secure data storage and transmission. A variety of solutions including asymmetric-key encryption schemes and symmetric-key encryption schemes have been proposed.

Encryption technologies can be regarded as primitive tools to implement access control in the EHR systems. Although there are a variety of methods which can be applied to access control, encryption technologies have played a main role at the lower level of the computer system to implement access control. For example, with a simple symmetric-key encryption, the encrypted data can only be accessed by the authorised user who holds the key. Similarly, in the asymmetric-key system, the public-key encryption only allows the authorised user who holds the private key to decrypt the encrypted data with the sender’s public key. It is noticed that asymmetric-key schemes are more expressive and powerful in construction of fine-grained access control systems. For example, data access control can be achieved through identity-based encryption (IBE) [Sha84] where public keys are identities of users, such as name and email address, and can be used as the subjects in an access control approach. It can be also applied to role-based access controls where the access rights are allocated to users based on their roles. With attribute-based encryption (ABE), we can achieve fine-grained sharing of encrypted data [GPSW06], which is even more expressive access control. Two different types of ABE have been proposed by Pandey et al. [GPSW06] and Bethencourt et al. [BSW07], respectively, including ciphertext policy attribute based encryption (CP-ABE), and key policy attribute based encryption (KP-ABE). The encryption schemes could be employed based on the system’s requirements, whether a policy based is desired or key based.

In IBE schemes, the receiver and sender in the encryption scenario have to obtain their private keys from a trusted third party (private key generator or PKG). The implementation of centralised trusted authorities in IBE has become a controversial issue, because they could become bottlenecks when many users are involved. ABE
schemes also inherit the issue, unfortunately. The potential solution is to utilise multiple authorities; therefore, the task of PKG can be handled by multiple authorities [Cha07, CC09]. Decentralised ABE approaches have shown potential applicability in EHRS, which is usually large with users who reside in different geographic locations.

Nonetheless, the implementation of cryptographic solutions in practice is challenging, in terms of their computational costs and technical complexities. Some medical devices might have limited computational power, and lightweight algorithms are required to their operations. For example, body sensor devices, which are used to monitor patient’s vital health status, are required to proceed quickly.

**Authentication:**

While encryption protects the data, it does not guarantee that the sender is genuine. Then, authentication as one of the most important security requirements for EHR systems assures the origin of the sender. Authentication could be addressed with several methods: (1) In a symmetric-key encryption scheme, the sender and the receiver can properly authenticate each other through a mutual authentication protocol. The idea is that the secret key shared between them is known to each party; therefore, the encrypted message indicates that the encryptor/sender is the legitimate user, while the receiver can correctly decrypt the message. When the encrypted response has been received from the receiver, the sender acknowledges that the receiver is legitimate. This approach of authentication has been widely applied in networked systems in order to build private communication protocols. (2) As a more efficient analogous scheme, Message Authentication Code (MAC) can provide a similar function, where the MAC key is shared by the sender and receiver only. The computation of a MAC is more efficient than an encryption. The MAC based authentication schemes have been widely adopted in computer networks such as IPSec. (3) Public-key cryptography can offer more powerful authentication schemes, especially with digital signatures. As a digital signature can only be generated by the signer who owns the private signing key, the verifier of the signature can be confirmed of legitimacy of the signer with the corresponding public key; therefore, the authenticity of the claimed sender and message can be confirmed. This also provides integrity properties, i.e. the sender can not deny its action in sending the data and more importantly the data has not been modified in transmission. In this thesis, we adopted all these approaches to build novel secure EHR systems to meet our goals.
1.2. Motivation of research

Privacy:

In EHR systems, privacy is ensured by protecting the patient identities and data from disclosing to unauthorised parties. There are various users in an EHRS; including doctors, specialists, nurses, pharmacists and patients who should be given different privileges for data access. This can be achieved through granting proper access rights to authorised users, restricting any unauthorised access to patient data. To protect data privacy, the usual access control mechanisms can help. That is, once the data is encrypted, only the authorised users can access it. However, to achieve identity privacy, we could utilise different approaches. Obviously, encryption of identity can help to protect identity privacy. This approach can be usually applied to database environment. In networked systems, the IP numbers or network addresses need to be protected as well, as they can be regarded as identification information of patients. In this case, we cannot apply traditional encryption schemes as computer networks require the IP numbers to deliver the network packets. Therefore, we need to handle this with different approaches. One of the solutions is to apply subliminal identities [MV96], or we could utilise MIX NET.

Distributed systems and security:

Since, in the real world, medical centres reside in various geographical locations, healthcare providers need to provide medical systems that offer services to patients and doctors in large-scale distributed systems. However, providing electronic health services to users who reside in remote or rural areas is challenging. This becomes more significant when accessing a specialist who mostly resides in a metropolitan area is inconvenient and costly. The medical systems, consequently, need to be designed to provide users mobility, in which remote users also can benefit from medical cares. Data sharing in distributed systems, however, has raised security and privacy concerns because multiple users can access the data. How to establish security and privacy protections while allowing remote patients to access doctors at any location is a research challenge.

In a distributed environment, patients are managed by different healthcare givers (e.g. hospitals) that handle users registrations and treatment procedures. Since a central authority can not handle data access for healthcare services, cloud computing as an effective solution enables ubiquitous and on-demand data access through sharing the computing resources. Such systems have both advantages and disadvantages to both users and healthcare providers. While users receive medical services
1.3 Thesis Contributions

While various security safeguards and privacy protections have been proposed in EHRS, there are still many research challenges to be solved. In order to investigate the current security gaps and provide effective solutions, we performed a literature review on security and privacy aspects of electronic health record (EHR) systems [RWS15], from technical perspectives. We have done an exhaustive search on related works of security and privacy implementations and current issues in electronic healthcare systems. We have identified the most important features that are significant in the implementation of electronic health record systems. As a result, access control policies, data sharing and cryptography techniques in the literature are highlighted as important aspects of security and privacy implementations in EHR systems.

Since electronic health records could be operated in large-scale distributed environments, handling the patient data access becomes significant. In these systems, implementing a central authority results in a bottleneck where there are many users in the system. We proposed a secure multi-authority framework that handles system users in a large-scale distributed environment [RMSW16]. Our solution controls the access to EHR data and improves the scalability and applicability of an EHR system in a distributed environment, such as cloud computing. Accordingly, we proposed our system model and attribute-based protocols that sketch the secure communication channels among users. We also presented an application scenario that shows the applicability of our solutions.

Security and privacy preservation are prominent when data is shared in a distributed system, and there are various domains with distinct access requirements. Therefore, an access control method needs to be defined for privacy protection of EHRs. Encryption techniques can be used in order to protect data content against any unauthorised modification, however, access control policies are needed to be transformed to adapt for the situation. We proposed an access control policy mechanism for privacy preservation and policy transformation of Electronic health records in hybrid clouds [RM16]. With our solution, an EHR access-control policy can be
transformed from a private cloud to a public cloud while it is encrypted. The transformed data remains confidential while it is re-encrypted and shared among authorized users. Our novel solution contributes to data sharing in EHR systems where handling different access control policies among different clouds seems unachievable.

In distributed systems, integration of access control policies is important to provide comprehensive care, but this is inevitable without access control policy combination mechanisms. How to combine policies to provide flexibility of access rights in which users’ features are dynamic, i.e., users’ locations and access rights are changing in dynamic systems, is challenging. We introduced an XACML access control policy combination mechanism which handles multiple access control policies through a similarity analysis phase [RM17a]. Our solution contributes to distributed systems where flexibility in access control policies is desired in system developments. According to our solution, different XACML policies are evaluated to decide whether they can be combined or not. The similarity analysis and combination algorithms are proposed to combine the heterogeneous access control policies and offer comprehensive services at a large-scale healthcare system. Our solution fosters interoperability and scalability among healthcare providers while preserving patient’s privacy and data security.

In recent years, with the adoption of telemedicine systems, it has become feasible to provide health services to remote patients. However, providing secure telemedicine systems which capture users mobility and privacy is still challenging. We proposed security protocols that capture patients and doctors mobility in telemedicine systems [RM17b]. Our solution contributes to telemedicine services in remote and rural areas where patients and doctors can establish consultation sessions at any time and location they reside. We presented different practical application scenarios which demonstrate communication sessions among users and the intermediary servers. We employed symmetric-key encryption in our protocols that assures security and privacy of data exchange. Our protocols achieve patient anonymity, data confidentiality, data integrity, freshness and mutual authentication features.

Patient monitoring applications have been widely adopted in healthcare systems. However, ensuring data security and patients privacy where efficiency has not adversely affected is under debate. In patient monitoring devices, patient’s health data (e.g. blood pressure, body temperature) is sent to the doctor for the evaluation of patient’s health status; however, the data needs to be protected against any
modification or lost. We proposed a provably secure homomorphic sign-encryption scheme for patient monitoring devices which ensures data integrity, confidentiality and authentication [RMZ17]. Our novel solution addresses IND-CPA and Weak Unforgeability (WUF) security goals under DDH and CDH assumptions, respectively. We contribute to patient monitoring devices in ensuring confidentiality, integrity and authentication where the collected data in a timely manner can be used by doctor to improve the patient’s health.

1.4 Thesis Organisation

The remainder of this thesis is organised as follows.

In chapter 2, we review the preliminaries used in this thesis. We introduce the complexity assumptions, security notions and algebra knowledge. In addition, we present some basic cryptographic primitives which are being used in our schemes, including hash function, random oracle model, public key encryption, sequences of games, digital signature and signcryption. We also review the preliminary of XACML structure.

In chapter 3, we develop a multi-authority security framework for scalable EHR systems in a distributed environment. We present the system models and security protocols that improve scalability in the medical systems.

In chapter 4, we design a dynamic access control transformation mechanism that enables the health information sharing in a distributed environment. We also present the system models along with the application scenario.

In chapter 5, we propose an access control policy combination scheme from similarity analysis in the secure and privacy-preserved EHR systems. We illustrate an access control policy combination framework in XACML and algorithms that evaluate different policies for similarities. We also give the instances of real case scenarios and access control policies in XACML.

In chapter 6, we construct practical and secure protocols for user mobility in telemedicine systems. We illustrate different application scenarios and relevant protocols. Our solutions enable secure and privacy preserved Telemedicine systems for remote patients.

In chapter 7, we propose a novel homomorphic sign-encryption scheme and illustrate its application in patient monitoring devices. We present the system model, security model and the security analysis of our scheme.
Finally, in chapter 8 we discuss our future research and conclude this thesis.
Chapter 2

Preliminaries

This thesis consists of security protocols, access control schemes, security policies and cryptographic schemes. To aid the reader, this chapter introduces related preliminaries and notions which have been used throughout this thesis, and illustrates the foundations of algebra, complexity assumptions and cryptographic tools. For more information regarding the cryptography foundations refer to [Mao04, Gol04, Sho06].

2.1 Abstract Algebra

Denote by \(\mathbb{Z} = \{\cdots, -2, -1, 0, 1, 2, \cdots\}\) the set of integers. \(\mathbb{Z}_p\) denotes the set of \(\{0, \cdots, p-1\}\) where \(p\) is a prime number and \(\mathbb{Z}_p^*\) is the set of positive integers smaller than \(p\) and relatively prime to \(p\),

\[
\mathbb{Z}_p^* = \{n|1 \leq n \leq p, \gcd(n, p) = 1\}.
\]

Denote by \(\lambda\) a security parameter and by \(1^\lambda\) the string of \(\lambda\) ones and denoted by \(r \in \mathbb{Z}_p\) the polynomial \(r\) which is randomly selected from the polynomial ring \(\mathbb{Z}_p\) consisting of the polynomials that coefficients are from the finite field \(\mathbb{Z}_p\).

We say that a function \(\epsilon : \mathbb{Z} \to \mathbb{R}\) is negligible if for all \(k \in \mathbb{Z}\), there exists \(N \in \mathbb{Z}\) such that \(\epsilon(n) \leq \frac{1}{n^k}\) for all \(n > N\). By \(\epsilon\), we always denote a negligible function that is asymptotically smaller than any inverse polynomial function.

For any \(a, b \in \mathbb{Z}\), we have \(\gcd(a, b)\) as the greatest common divisor of \(a\) and \(b\). We say that \(a, b \in \mathbb{Z}\) are “relatively prime” if \(\gcd(a, b) = 1\); otherwise \(a, b \in \mathbb{Z}\) are “composite”.

2.2 Foundations of Algebra

In this section, we review the basic algebra knowledge, including field, group and cyclic group.
2.2.1 Field

Field consists of a set of elements and two operations defined between any two elements in the set.

Definition 2.1 (Field) A field \((F, +, \circ)\) consists of a set \(F\) and two operations: addition \(\oplus\) and multiplication \(\circ\), and satisfies the following properties,

- **Addition Group.** \((F, +)\) is an Abelian group. The identity of the group \((F, +)\) is denoted as \(0_F\) (additive identity or zero-element);
- **Multiplication Group.** Let \(F^* = F - \{0_F\}\). \((F^*, \circ)\) is an Abelian group. The identity of the group \((F^*, \circ)\) is denoted as \(1_F\) (multiplicative identity);
- **Distributivity.** \(\forall a, b, c \in F, (a \oplus b) \circ c = (a \circ c) \oplus (b \circ c)\).

2.2.2 Group

A group \(G\) is a set of elements with a binary operation \(\circ\) which is executed between any two elements in the set. If \(a, b \in G\), then we write \(a \circ b\).

Definition 2.2 (Group) A group \((G, \circ)\) is a non-empty set \(G\) equipped with a binary operation \(\circ\) over elements in \(G\), satisfying the following properties,

- **Closure.** \(\forall a, b \in G\), then \(a \circ b \in G\).
- **Associativity.** \(\forall a, b, c \in G\), such that \((a \circ b) \circ c = a \circ (b \circ c)\).
- **Existence of an identity.** \(\exists e_G \in G, \forall b \in G\), such that \(e_G \circ b = b \circ e_G = b \in G\).
- **Existence of inverses.** \(\forall a \in G\), \(\exists a^{-1} \in G\), such that \(a \circ a^{-1} = a^{-1} \circ a = 1_G\).

For simplicity, a group \((G, \circ)\) is often denoted as \(G\) when the operation \(\circ\) is clear in the context. The number of the elements in \(G\) is called the order of \(G\) and denoted as \(n = |G|\) where \(g \in G, g^n = 1\) (1 denote to the identity of group \(G\)). A group \(G\) is a finite group if \(|G|\) is finite; otherwise, it is an infinite group. Let \(G(1^3)\) be a group generator which takes as input \(1^3\) and outputs a group \(G\) with order \(p\), namely \(G(1^3) \rightarrow (p, G)\).

If the binary operation \(\circ\) is an addition “+”, then the group is called an “additive group” where the identity element is denoted by \(0_G\) (Zero element), and the inverse
of an element \( a \in G \) is denoted by \(-a\). If the binary operation \( \circ \) is a multiplication 
\( \cdot \), then the group is called a multiplicative group where the identity element is
denoted by \( 1_G \), and the inverse of an element \( a \in G \) is denoted by \( a^{-1} \).

We say a group \((G, \circ)\) is an Abelian group if it has the following property,

- Commutivity. \( \forall a, b \in G, a \circ b = b \circ a. \)

**Definition 2.3 (Order of Group Element)** Suppose that \( g \in G \), the order of \( g \) in \( G \)
is the least \( i \in \mathbb{Z}^+ \) such that \( g^i = 1_G \). If for all \( i \in \mathbb{Z}^+ \), \( g^i \neq 1_G \), the order of \( g \) is infinite. The order of \( g \) is denoted as \( \text{ord}(g) = |G| \).

Especially, if any element in a group \( G \) can be expressed by a specially element in \( G \), \( G \) is called as a cyclic group. The formal definition of a cyclic group is as follows:

**Definition 2.4 (Cyclic Group)** A group \( G \) is a cyclic group if there exists \( g \in G \) such that \( G = \{g^i | i \in \mathbb{Z}\} \). The element \( g \) is called a generator of the group \( G \). \( G \) is said to be generated by \( g \) and represented as \( G = \langle g \rangle \).

**Definition 2.5 (Isomorphism)** Let \( G, G' \) be groups wrt the operations \( \circ_G, \circ_{G'} \), respectively. A function \( \rho : G \rightarrow G' \) is an isomorphism from \( G \) to \( G' \) if,

- \( \rho \) is a bijection,
- \( \forall a, b \in G \) then \( \rho(a \circ_G b) = \rho(a) \circ_{G'} \rho(b) \).

### 2.3 Bilinear Groups

In this section, we review the knowledge related to bilinear groups.

**Definition 2.6 (Bilinear Map on Prime Order Groups)** [BF03]. Let \( G_1, G_2 \) be multiplicative cyclic groups of prime order \( p \). Let \( G_\tau \) be a multiplicative group of prime order \( p \). Let \( g_1 \) and \( g_2 \) be the generators of \( G_1 \) and \( G_2 \), respectively. A bilinear map (pairing) \( e : G_1 \times G_2 \rightarrow G_\tau \) satisfies the following properties,

1. Bilinearity. \( \forall x \in G_1, y \in G_2 \) and \( a, b \in \mathbb{Z}_p \), \( e(x^a, y^b) = e(x, y)^{ab} \).
2. Non-degeneracy. \( e(g_1, g_2) \neq 1_{G_\tau} \) where \( 1_{G_\tau} \) is the identity element in \( G_\tau \).
3. **Computability.** For all \( a \in \mathbb{G}_1, b \in \mathbb{G}_2 \), there exists an efficient algorithm to compute \( e(a, b) \).

**Definition 2.7 (Bilinear Map)** [BF03]. Let \( \mathbb{G}_1, \mathbb{G}_2 \) be cyclic groups of order \( n \), where \( n = pq \) and \( p, q \) be two prime numbers. Let \( g_1 \) be the generator of \( \mathbb{G}_1 \) and \( g_2 \) be the generator of \( \mathbb{G}_2 \). A bilinear map (pairing) \( e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T \) on composite order groups satisfies the following properties,

1. **Bilinearity.** \( \forall g_1 \in \mathbb{G}_1, \forall g_2 \in \mathbb{G}_2 \) and \( \forall a, b \in \mathbb{Z}_p \), \( e(g_1^a, g_2^b) = e(g_1, g_2)^{ab} \).
2. **Non-degeneracy.** \( e(g_1, g_2) \neq 1_{\mathbb{G}_T} \) where \( 1_{\mathbb{G}_T} \) is the identity element in \( \mathbb{G}_T \).

**Definition 2.8 (Bilinear Groups)** [GPS08]. \( \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T \) construct a bilinear group if there exists a bilinear map \( e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T \), where \( |\mathbb{G}_1| = |\mathbb{G}_2| = |\mathbb{G}_T| = p \).

Galbraith et al. [GPS08] divided pairing operations used in cryptography into three types:

- **Type 1:** \( \mathbb{G}_1 = \mathbb{G}_2 \);
- **Type 2:** \( \mathbb{G}_1 \neq \mathbb{G}_2 \), there exists an efficiently computable homomorphism map \( \psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1 \);
- **Type 3:** \( \mathbb{G}_1 \neq \mathbb{G}_2 \), there are no efficiently computable homomorphism maps between groups \( \mathbb{G}_1 \) and \( \mathbb{G}_2 \).

When \( \mathbb{G}_1 = \mathbb{G}_2 \), the pairing is symmetric (there exists efficiently computable isomorphism between the two groups) and we denote the symmetric bilinear group as \( (e, p, \mathbb{G}_1, \mathbb{G}_T) \). However, when \( \mathbb{G}_1 \neq \mathbb{G}_2 \) the pairings are asymmetric. Since, pairing is often constructed on elliptic curves, \( \mathbb{G}_1 \) and \( \mathbb{G}_2 \) are subgroups of the group of points on an elliptic curve, its efficiency is dependent on the selected elliptic curves over the finite fields.

### 2.4 Complexity Assumptions

In this section, we review the complexity assumptions used throughout this thesis.

**Definition 2.9 (Discrete Logarithm (DL) Assumption)** [Odl84]. Let \( \mathbb{G} = \langle g \rangle \) be a cyclic group of prime order \( p \) generated by a generator \( g \). Given \( g \in \mathbb{G} \), we say
that the discrete logarithm assumption holds on $G$ if no PPT adversary $A$ can compute $g^a$ such that $a \in_R \mathbb{Z}_p$ with the the advantage

$$\text{Adv}^{\text{DL}}_A = \Pr[g^a \leftarrow A(p, g, G)] \geq \epsilon(\lambda)$$

where the probability is taken over the random choice of $g^b \in G$ and the bits consumed by the adversary $A$.

The DL assumption assumes that the advantage $\text{Adv}^{\text{DL}}_A$ is negligible for any probabilistic polynomial time (PPT) algorithm $A$ under the security parameter $1^\lambda$.

**Definition 2.10 (Computational Diffie-Hellman (CDH) Assumption) [DH76].** Let $G = \langle g \rangle$ be a cyclic group of prime order $p$ generated by a generator $g$. Given $g, g^a, g^b \in G$ for randomly selected $a, b \in_R \mathbb{Z}_p$, there exists an algorithm $A$ that computes $g^{ab}$ with the advantage

$$\text{Adv}^{\text{CDH}}_A = \Pr[g^{ab} \leftarrow A(G, p, g, g^a, g^b) \mid a, b \in_R \mathbb{Z}_p, G = \langle g \rangle].$$

The CDH assumption assumes that the advantage $\text{Adv}^{\text{CDH}}_A$ is negligible for any PPT algorithm $A$ under the security parameter $1^\lambda$.

**Definition 2.11 (Decisional Diffie-Hellman (DDH) Assumption) [Bon98].** Let $G = \langle g \rangle$ and $a, b \in_R \mathbb{Z}_p$ as described in the CDH assumption. Given $g, g^a, g^b$, there exists an algorithm $A$ that distinguishes $g^{ab}$ with a random element $Z \in_R G$ with the advantage

$$\text{Adv}^{\text{DDH}}_A = \left| \Pr[1 \leftarrow A(G, p, g, g^a, g^b) \mid g^{ab}] - \Pr[1 \leftarrow A(G, p, g, g^a, g^b, Z) \mid Z \in_R G] \right|$$

The DDH assumption assumes that the advantage $\text{Adv}^{\text{DDH}}_A$ is negligible for any PPT algorithm $A$ under the security parameter $1^\lambda$.

### 2.5 Cryptographic Tools

In this section, we introduce some useful cryptographic tools, including hash function, random oracle model, public-key encryption, sequences of games, digital signature, signcryption and an overview of XACML.
2.5. Cryptographic Tools

2.5.1 Hash Function

The universal classes of hash functions were introduced by Carter and Wegman [CW79]. A hash function $H : \{0, 1\}^* \rightarrow \{0, 1\}^n$ is a deterministic function that takes as input an arbitrary length string and outputs a fixed length string. A hash function offers the following properties [Mao04],

- **Pre-image resistance.** Given a hash function $H : \mathbb{X} \rightarrow \mathbb{Y}$ and value $y$, it is computationally infeasible to find a value $x \in \mathbb{X}$ such that $H(x) = y$;

- **Second pre-image resistance.** Given a hash function $H : \mathbb{X} \rightarrow \mathbb{Y}$ and value $x \in \mathbb{X}$, there is no efficient mechanism to find $x' \in \mathbb{X}$ such that $x \neq x'$ and $H(x) = H(x')$;

- **Collusion resistance.** Given a hash function $H : \mathbb{X} \rightarrow \mathbb{Y}$, it is computationally infeasible to find $x, x' \in \mathbb{X}$ such that $x \neq x'$ and $H(x) = H(x')$.

Hash functions with pre-image resistance and second pre-image resistance properties are denoted as “One-Way Hash Functions (OWHF)”. The hash functions with second pre-image resistance and collision resistance are denoted as “Collision Resistance Hash Functions”.

Hash functions play an important role as cryptographic primitives and building blocks to create encryption schemes and provide integrity in digital signatures, encryption, key agreement protocols, Message Authentication Code (MAC) [BCK96], etc.

Message Authentication Code (MAC), introduced by Carter and Wegman [CW79], is known as “Tag”. MAC is a short piece of information that added to the message for its authentication, i.e. to confirm the authenticity of the sender. Since computing a MAC require a secret key, it ensures data integrity as well as authenticity of the message.

2.5.2 Random Oracle Model

The random oracle model was introduced by Bellare and Rogaway [BR93] where a hash function is treated as a black box and the computation of hash value is possible through oracle queries. A random oracle is a hash function that responds to every query with a truly random space. By theory, the output of a hash function (random oracle) is computationally indistinguishable from the uniform distribution over its
output space. A random oracle combines the properties of deterministic, efficient and uniform output. In this model, an entity called “Simulator” simulates each party’s behaviour, in which that party makes a random oracle query to the random oracle $H$ in order to obtain an output on a specific value $x$. Then, the simulator $S$, which contains an $H$-table, checks the pairs $(x, H(x))$ in the $H$-table. If there exists such a query, the $S$ returns the value $H(x)$; otherwise, $S$ creates a new value $(x, H(x))$ uniformly at random from the output space of $H$ and stores new value in the current table.

Random oracle models are efficient tools to prove the security of schemes, in which we can construct a reduction of the action of an adversary $A$ in breaking the security of the scheme. Overall, protocols designed in this model are more efficient than those in standard model. However, as stated in [CGH04] a scheme which is proven to be secure in random oracle model is not necessarily secure in the standard model; unless otherwise mentioned so.

### 2.5.3 Public-Key Encryption

Public key cryptography (PKC) was introduced by Diffie and Hellman [DH76] in which two parties, sender and receiver, can communicate over a public communication channel without threatening the security of the system.

A public key encryption (PKE) is an asymmetric scheme that issues two different keys as secret key $SK$ and public key $PK$ to each party, as encryption key and decryption key. This is opposed to symmetric encryption schemes that decryption key and encryption key are the same. We can define a PKE scheme $\mathcal{PKE} = (\text{KeyGen}, \text{Enc}, \text{Dec})$ as following,

- **Setup**($1^\lambda$) $\rightarrow$ params. Taking a security parameter $1^\lambda$ as input, it outputs the system public parameters $\text{params}$.

- **KeyGen**($\text{params}$) $\rightarrow$ $(SK, PK)$. Taking the public parameters $\text{params}$ as input, it outputs an encryption (public) key and decryption (private) key pair $(PK, SK)$.

- **Enc**($\text{params}, PK, M$) $\rightarrow$ $C$. Taking the public parameters $\text{params}$, public key $PK$ and plaintext message $M$ as input, it outputs a ciphertext $C$.

- **Dec**($\text{params}, SK, C$) $\rightarrow$ $M$. Taking the public parameters $\text{params}$, private key $SK$ and ciphertext $C$, it outputs plaintext message $M$ if the ciphertext is valid; otherwise, outputs ⊥.
2.5. Cryptographic Tools

Definition 2.12 (Correctness) A public key encryption scheme is correct if the following equation holds:

\[
\Pr\left[\begin{array}{l}
\text{Dec}(\text{params}, SK, C) \rightarrow M \\
\text{KeyGen}(\text{params}) \rightarrow (SK, PK); \\
\text{Enc}(\text{params}, PK, M) \rightarrow C
\end{array}\right] = 1
\]

In public key encryption, there are three different types of adversaries, namely, Chosen-Plaintext Attack (CPA), non-adaptive Chosen-Ciphertext Attack (CCA) and adaptive Chosen-Ciphertext Attack (CCA2). In CPA, an adversary has access to an encryption oracle where can query any message of its choice and receive the corresponding encryption. In CCA, which is a stronger model than CPA, an adversary has access to a decryption oracle and can query any ciphertext of its choice to receive its corresponding plaintext. However, in CCA2, compared with CCA, an adversary can query to decryption oracle after receiving the challenge ciphertext, with the restriction that it can not send the challenge ciphertext again.

We can define a public key encryption scheme \( PKE = (\text{KeyGen}, \text{Enc}, \text{Dec}) \) security models as following.

Definition 2.13 (IND-CPA) A public key encryption scheme \( PKE = (\text{KeyGen}, \text{Enc}, \text{Dec}) \) is indistinguishable against adaptive chosen plaintext attacks (IND-CPA) if no PPT adversary \( \mathcal{A} \) by accessing to encryption oracle can win the game with the advantage,

\[
\text{Adv}^{\text{IND-CPA}}_{\mathcal{A}} = \Pr[b' = b] - \frac{1}{2} \geq \epsilon(\lambda).
\]

Setup. Simulator \( \mathcal{S} \) runs \( \text{Setup}(\text{params}) \) to generate the public parameters \( \text{params} \) and sends them to adversary \( \mathcal{A} \).

KeyGen. \( \mathcal{S} \) runs \( \text{KeyGen}(\text{params}) \) to generate the secret key and public key pair \((SK, PK)\) and sends the public parameters \( \text{params} \) with the public key \( PK \) to \( \mathcal{A} \).

Challenge. \( \mathcal{A} \) submits two messages \( m_0, m_1 \in M \), \( M \in \{0,1\} \) with equal length.

\( \mathcal{S} \) randomly selects \( m_b \) and computes \( C^* = \text{Enc}(\text{params}, PK, m_b) \), where \( b \in \{0,1\} \). \( \mathcal{S} \) responds \( \mathcal{A} \) with \( C^* \).

Guess. \( \mathcal{A} \) outputs its guess \( b' \) on \( b \in \{0,1\} \). \( \mathcal{A} \) wins the game if \( b' = b \).
Definition 2.14 (IND-CCA) A public key encryption scheme $\mathcal{PKE} = (\text{KeyGen}, \text{Enc}, \text{Dec})$ is indistinguishable against chosen ciphertext attacks (IND-CCA) if no PPT adversary $A$ making at most $q$-decryption queries can win the game with the advantage,

$$\text{Adv}_{A}^{\text{IND-CCA}} = \left| \Pr[b' = b] - \frac{1}{2} \right| \geq \epsilon(1^\lambda).$$

**Setup.** Simulator $S$ runs $\text{Setup}(1^\lambda)$ to generate the public parameters $\text{params}$ and sends them to adversary $A$.

**KeyGen.** $S$ runs $\text{KeyGen}(\text{params})$ to generate the secret key and public key pair $(SK, PK)$ and sends the public parameters $\text{params}$ with the public key $PK$ to $A$.

**Phase 1.** $A$ adaptively queries the decryption oracle. $A$ submits a ciphertext $C$ to $S$, where $C = \text{Enc}(\text{param}, PK, m)$. $S$ runs $\text{Dec}(\text{params}, SK, C)$ and responds $A$ with $m$. This query can be made multiple times.

**Challenge.** $A$ submits two messages $m_0, m_1 \in M$, $M \in \{0, 1\}$ with equal length. $S$ randomly selects $m_b$ and computes $C^* = \text{Enc}(\text{params}, PK, m_b)$, where $b \in \{0, 1\}$. $S$ responds $A$ with $C^*$.

**Phase 2.** $A$ adaptively query the decryption oracle. $A$ submits a ciphertext $C$ to $S$, where $C \neq C^*$. This query can be made multiple times.

**Guess.** $A$ outputs its guess $b'$ on $b \in \{0, 1\}$. $A$ wins the game if $b' = b$.

Definition 2.15 (IND-CCA2) A public key encryption scheme $\mathcal{PKE} = (\text{KeyGen}, \text{Enc}, \text{Dec})$ is indistinguishable against adaptive chosen ciphertext attacks (IND-CCA2) if no PPT adversary $A$ making at most $q$-decryption queries can win the game with the advantage,

$$\text{Adv}_{A}^{\text{IND-CCA2}} = \left| \Pr[b' = b] - \frac{1}{2} \right| \geq \epsilon(1^\lambda).$$

2.5.4 Sequences of Games

The sequences of games or “Game-Hopping” is a tool for taming the complexity in security proofs while the security proofs might become complicated and messy. Generally, there are three types of game hop: bridging steps, transitions based on indistinguishability and transitions based on (small) failure events.
2.5. Cryptographic Tools

**Bridging steps.** This form of game hop represents the ground of the game from the adversary $\mathcal{A}$’s point of view. There is no change in the environment and makes the $\mathcal{A}$’s probability of breaking the scheme as $\Pr[E_i] = \Pr[E_{i+1}]$.

**Transitions based on indistinguishability.** In this game hop a small change between two games, if detected by $\mathcal{A}$, can lead to an efficient method to distinguish among two distributions that are indistinguishable. In that way, we can construct an algorithm $\mathcal{D}$ that interpolates among two successive games, and evaluates the difference among two games. The indistinguishability assumption should imply that $\Pr[E_i] - \Pr[E_{i+1}]$ is negligible.

**Transitions based on (small) failure events.** In this type of transition, one can argue that two successive games proceed identically unless one event failed, then we can construct a difference lemma that computes $\Pr[E_i] - \Pr[E_{i+1}]$.

In order to construct the sequences of games proposed by Victor Shoup [Sho04], we need to proceed as follows,

1. **Construct.** One constructs a sequences of games, Game 0, Game 1, ..., Game $n$, where Game 0 is the original attack game with respect to adversary $\mathcal{A}$ and cryptographic primitive;

2. **Define.** Let $E_0$ be the event for $i = \{1, \cdots, n\}$ that defines an event $E_i$ in Game $i$ related to the definition of $E$;

3. **Proof.** The $\Pr[E_i]$ is negligibly close to $\Pr[E_{i+1}]$ for $i = \{0, \cdots, n-1\}$, and that $\Pr[E_n]$ is equal to (or negligibly close) the “Target Probability”.

In constructing proofs, the changes among successive games should be very small, in which the changes can be possible. Then, through one transition, if an adversary is successful to detect a small change, we would imply an efficient method of distinguishing between two indistinguishable distributions. Then, to prove that $|\Pr[E_i] - \Pr[E_{i+1}]|$ is negligible, we need to construct a distinguishing algorithm $\mathcal{D}$ that interpolates between two games Game $i$ and Game $i+1$ with a distinguisher algorithm $\mathcal{A}$.

Basically, one can design a “hybrid” game which is the combination of the two games and takes an auxiliary input. That means, one get Game $i$ if the auxiliary input is derived from $\mathcal{D}_1$ and get Game $i+1$ if it is driven from $\mathcal{D}_2$. Therefore,
the constructed distinguisher algorithm \( D \) runs the hybrid game with its input and outputs 1 if the relevant event occurs.

### 2.5.5 Digital Signature

The digital signature scheme proposed by Diffie and Hellman [DH76] is the electronic format of a handwritten signature. A digital signature offers useful security properties, including integrity, authentication and non-repudiation. Integrity means that data has not been modified during transmission. Authentication convinces the receiver that the data is sent from the authorized sender. Non-repudiation indicates that the user cannot deny its action on the data. In fact, a valid digital signature can convince the verifier that the data is generated by the authorized party for the public access.

A digital signature scheme consists of the following four algorithms [GJKR96],

\[
\text{Setup}(1^\lambda) \rightarrow \text{params}. \text{ The setup algorithm takes as input security parameter } 1^\lambda \text{ and outputs the public parameters } \text{params}. \\
\text{KeyGen}(\text{params}) \rightarrow (SK, PK). \text{ The key generation algorithm takes as input security parameter } 1^\lambda \text{ and outputs a secret key and public key pair } (SK, PK). \\
\text{Sign}(\text{params}, SK, M) \rightarrow \sigma(M). \text{ The signature algorithm takes as input the public parameters } \text{params}, \text{ the secret key } SK \text{ and a message } M, \text{ and outputs a signature } \sigma(M) \text{ on message } M. \\
\text{Verify}(\text{params}, M, PK, \sigma) \rightarrow 1/0. \text{ The verification algorithm takes as input the public parameters } \text{params}, \text{ the message } M, \text{ the public key } PK \text{ and the signature } \sigma(M), \text{ and outputs 1 if } \text{Sign}(\text{params}, M, SK) \rightarrow \sigma(M); \text{ otherwise, it outputs 0.}
\]

**Definition 2.16 (Correctness)** The correctness of the digital signature is represented as following,

\[
\Pr \begin{bmatrix}
\text{Verify}(\text{params}, M, PK, \sigma) \rightarrow 1 \\
\text{Setup}(1^\lambda) \rightarrow \text{params}; \\
\text{KeyGen}(\text{params}) \rightarrow (SK, PK); \\
\text{Sign}(\text{params}, SK, M) \rightarrow \sigma.
\end{bmatrix} = 1
\]

A digital signature scheme should basically achieve the security notion of *existential unforgeability under adaptive chosen message attack* (EU-CMA) [GMR88]. The
EU-CMA secure scheme is defined as the following game that is executed between a simulator $S$ and adversary $\mathcal{A}$,

**Setup.** $S$ runs $\text{Setup}(1^\lambda)$ algorithm to output the public parameters $\text{params}$ and then sends them to $\mathcal{A}$.

**KeyGen.** $S$ runs $\text{KeyGen}(\text{params})$ algorithm to output a secret and public key pair $(SK, PK)$ and sends $PK$ to $\mathcal{A}$.

**Query.** $\mathcal{A}$ adaptively queries the signature oracle.
- $\mathcal{A}$ sends a message $M$ to $S$. $S$ runs $\text{Sign}(\text{params}, SK, M)$ algorithm to generate a signature $\sigma(M)$ and responds $\mathcal{A}$ with $\sigma$. The query can be repeated many times.

**Forgery.** $\mathcal{A}$ outputs a message and signature pair $(M^*, \sigma^*)$.
- $\mathcal{A}$ wins the game if $M^*$ has not been queried in the signature oracle, and $\text{Verify}(\text{params}, M^*, PK, \sigma^*) \rightarrow 1$.

**Definition 2.17 (EU-CMA)** A digital signature scheme is $(\tau, q, \epsilon(1^\lambda))$-existentially unforgeable against adaptive chosen message attacks (EU-CMA) if no probability polynomial time (PPT) adversary $\mathcal{A}$ can win the game with the advantage

$$\text{Adv}^{\text{EU-CMA}}_\mathcal{A} = \text{Pr}[\text{Verify}(\text{params}, M^*, PK, \sigma^*) \rightarrow 1] \geq \epsilon(1^\lambda)$$

in the above security model.

There is a stronger notion of EU-CMA that is called strongly existential unforgeability under an adaptive chosen message attack (SEU-CMA) [ADR02], as following game,

**Setup.** $S$ runs $\text{Setup}(1^\lambda)$ to output the public parameters $\text{params}$ and sends them to $\mathcal{A}$.

**KeyGen.** $S$ runs $\text{KeyGen}(\text{params})$ to output a secret and public key pair $(SK, PK)$ and sends $PK$ to $\mathcal{A}$.

**Query.** $\mathcal{A}$ adaptively query the signature oracle. $\mathcal{A}$ adaptively sends messages $\{M_1, M_2, \cdots, M_q\}$ to $S$. $S$ runs $\text{Sign}(\text{params}, SK, M_i)$ to generate a signature $\sigma_i(M_i)$ and responds $\mathcal{A}$ with $\sigma_i$, for $i = \{1, 2, \cdots, q\}$. 
Forgery. $A$ outputs a message and signature pair $(M^*, \sigma^*)$. $A$ wins the game if
$(M^*, \sigma^*)$ has not been queried before $(M^*, \sigma^*) \notin \{(M_1, \sigma_1), (M_2, \sigma_2), \ldots, (M_q, \sigma_q)\}$
and $\text{Verify}(\text{params}, M^*, PK, \sigma^*) \rightarrow 1$.

**Definition 2.18 (SEU-CMA)** A digital signature scheme is $(\tau, \epsilon(1^\lambda))$-strongly
existentially unforgeable against adaptive chosen message attacks (SEU-CMA) if no
PPT adversary $A$ can win the game with the advantage

$$\text{Adv}^\text{SEU-CMA}_A = \Pr[\text{Verify}(\text{params}, M^*, PK, \sigma^*) \rightarrow 1] \geq \epsilon(1^\lambda)$$

in the above security model.

### 2.5.6 Signcryption

A signcryption scheme [Yul97] consists of probabilistic polynomial time algorithms
($\text{Setup}$, $\text{KeyGen}$, $\text{Signcrypt}$, $\text{DeSigncrypt}$, $\text{Verify}$).

$\text{Setup}(1^\lambda) \rightarrow \text{params}$. The setup algorithm takes as input security parameter $1^\lambda$
and outputs the system public parameters $\text{params}$.

$\text{KeyGen}(\text{params}) \rightarrow (SK_s, PK_s)$. The key generation algorithm takes as input the
system public parameters $\text{params}$ and outputs a secret key and public key
pair $(SK_s, PK_s)$ of the sender.

$\text{KeyGen}(\text{params}) \rightarrow (SK_r, PK_r)$. The key generation algorithm takes as input the
system public parameters $\text{params}$ and outputs a secret key and public key
pair $(SK_r, PK_r)$ of the receiver.

$\text{Signcrypt}(\text{params}, SK_s, PK_r, M) \rightarrow SC(M)$. The signcryption algorithm takes as
input the public parameters $\text{params}$, the secret key of the sender $SK_s$, the
public key of the receiver $PK_r$ and a message $M$, and outputs a signcryption
$SC(M)$ on message $M$.

$\text{DeSigncrypt}(\text{params}, SK_r, PK_s, SC(M)) \rightarrow M$. The signcryption algorithm takes as
input the public parameters $\text{params}$, the secret key of the receiver $SK_r$, the
public key of the sender $PK_s$ and a sigcrypted message $SC(M)$, and outputs
the plaintext message $M$. 

Verify\( (\text{params, SC(M), M’, PK_s, SK_r}) \rightarrow 1/0 \). The verification algorithm takes as input the public parameters \( \text{params} \), the signcrypted message \( \text{SC}(M) \), the designcrypted message \( M_0 \), the sender’s public key \( PK_s \) and the receiver’s private key \( SK_r \), and outputs 1 if \( M = M’ \); otherwise, it outputs 0.

**Definition 2.19 (Completeness)**  For any \( M \), \( \text{KeyGen}(1^\lambda) \rightarrow (SK_s, PK_s) \) and \( \text{KeyGen}(1^\lambda) \rightarrow (SK_r, PK_r) \) such that \( SK_s \neq SK_r \), a signcryption \( SC \) is complete if the following holds,

\[
\text{DeSigncrypt}(\text{SC}(M, SK_s, PK_r), SK_r) \rightarrow M,
\]

\[
\text{Verify}(M_0, \text{SC}(m, SK_r, PK_s)) \rightarrow 1.
\]

The security models of signcryption that have been employed in this thesis are confidentiality and unforgeability, as follows.

**Definition 2.20 (Confidentiality)**  A signcryption scheme is semantically secure against chosen plaintext attacks (IND-CPA) if no PPT adversary has a non-negligible advantage in the following game,

1. **Setup.** Simulator \( S \) runs \( \text{Setup}(1^\lambda) \) to generate the public parameters \( \text{params} \) and sends them to adversary \( A \).
2. **KeyGen.** \( S \) runs \( \text{KeyGen}(\text{params}) \) to generate the secret key and public key pair for the receiver \( (SK_r, PK_r) \) and sends the public parameters \( \text{params} \) with the public key of the receiver \( PK_r \) to \( A \).
3. **Challenge.** \( A \) submits two messages \( m_0, m_1 \in M \) of equal length where \( M \in \{0, 1\} \) and the secret key of the sender \( SK_s \). If \( PK_r \neq PK_s \) and \( PK_r \) is valid then \( S \) randomly selects \( m_b \) and computes \( \text{SC}^* \leftarrow \text{Signcrypt}(\text{params, PK_r, SK_s, m_b}) \), where \( b \in \{0, 1\} \). \( S \) responds \( A \) with \( \text{SC}^* \). Otherwise, it returns \( \bot \).
4. **Guess.** \( A \) outputs its guess \( b’ \) on \( b \in \{0, 1\} \). \( A \) wins the game if \( b’ = b \).

\( A \)'s advantage in winning the above game is defined as

\[
\text{Adv}_{\text{IND-CPA}}^A = \left| \Pr[b = b’] - 1/2 \right|.
\]

**Definition 2.21 (Unforgeability)**  A signcryption scheme is existentially unforgeable against chosen-message attacks (EU-CMA) if no PPT adversary \( A \) has a non-negligible advantage in the following game,
2.6. An overview of XACML

XACML is an OASIS standard language as a basis to provide a common security policy language for different platforms and tools with pre-defined syntax and semantics [OAS13]. It supports a flexible data management and access tool. Access was defined by XACML standard as “performing an action” and access control defined as “controlling access in accordance with a policy or policy-set”.

The policy model in XACML consists of three preliminary components; Policy-set, Policy, Rule, as shown in Fig. 2.1. PolicySet components are associated with Target, Policy and Policy Combining Algorithm (PCA). The PCA specifies the process which the results of evaluating the policies are combined. Policy is also associated with Target, Rule Combining Algorithm (RCA) and Rule. Rule consists of one or multiple rules and is associated with Target, Condition and Effect. Rule in a policy is the most primarily unit, which is encapsulated within policy. Rule indicates the set of requests to which it applies, which is identified by Subject, Resource, Action, Environment attributes, in the form of logic. The Condition element refines the applicability of the target. The Effect of the rule identifies the intended consequence of a rule, which consists of Permit and Deny. All these components are used to test whether a request is applicable to the policy or not; can be specified in the condition section.

Fig. 2.2 depicts the XACML data-flow with the main actors in the XACML domain including: Context Handler, Policy Enforcement Point (PEP), Policy Decision Point (PDP), Policy Information Point (PIP) and Policy Administration Point.
2.6. An overview of XACML

(PAP). PAP makes policies available to PDP when the access requester sends a request to PEP to access the policies. Then, PEP sends the request to context handler, and context handler forwards this request to PDP.

With respect to a particular decision request, PDP evaluates a policy-combining algorithm (PCA) or rule-combining algorithm (RCA) and a set of policies, policy-sets or rules. It returns a response context that specifies the authorisation decision with Decision element of “Permit”, “Deny”, “Indeterminate” or “NotApplicable” values.

Due to the need of attributes information, PDP sends the request to the context handler, in which case context handler forwards this request to PIP. Next, the
context handler sends the retrieved attributes in addition to resource attributes to PDP. PDP then returns the response context to the context handler that translates it to the native response format of the PEP. Finally, PEP fulfils the decisions, whether it is deny or permit.
Chapter 3

Multi-Authority Security Framework for Scalable EHR Systems

Electronic Health Record (EHR) systems can be operated in a large-scale distributed environment, such as cloud computing, which might have to be managed by multiple authorities who control the access to patient records. Unfortunately, the security of such systems is usually inadequate, which results in the hindrance of the EHR systems adoption in practice. Attribute-based systems have been a popular choice that could provide a flexible and reliable access control to EHR databases, which are usually managed by a single authority, who is responsible for setting up the system’s policy. In a large-scale distributed system, it might be necessary to have multiple authorities, who can handle users located at different areas. Nevertheless, one of the challenges is how to enable multiple authorities with a single access policy. In this chapter, we provide a sound solution to this issue. Our EHR system provides a secure environment for EHR users to use the system conveniently and provide the flexibility and scalability.

3.1 Introduction

The Electronic Health Record Committee in the Health Information Management Systems Society (HIMSS) defined EHR using this statement: “The Electronic Health Record (EHR) is a secure, real-time, point-of-care, patient-centric information resource for clinicians. The EHR aids clinicians’ decision making by providing access to patient health record information where and when they need it by incorporating evidence-based decision support” [DL14]. The 2003 ISO/TS 18308 references the IOM 1991 definition and CEN 13606 2000 to define an EHR system as “a system for recording, retrieving and manipulating information in electronic health records” [DFH04].
EHR has numerous advantages for improving the quality of diagnosis and reducing the medical costs and errors in order to address reliable and efficient healthcare processes. The exchange of health information is a crucial component to enable provision of high-quality health services. Meanwhile, one of the key issues in electronic healthcare is to share patient records across enterprises. Healthcare providers require to share and distribute EHR data among necessarily interested parties to provide access to healthcare resources and achieve EHR advantages.

Deploying cloud services in the health sector can facilitate the exchange of medical data among entities and act as the medical record storage but require some certain level of protections [AK14]. In this regard, several regulations and standards such as HITECH Act, HIPAA, HL7 CDA, CEN 13606 EHRcom and openEHR proposed guidelines and frameworks for sharing and exchanging health information via digitally representation of clinical data between different entities across healthcare communities. In the following section, we outline some Health IT standards proposed for healthcare data exchange and data sharing.

### 3.1.1 Health IT Standards

Health Level-7 (HL7) is an acceptable messaging standard that refers to a set of flexible international standards, guidelines and methodologies for clinical and administrative data exchange among software applications used by various healthcare providers. These standards can ensure healthcare system interoperability and EHR sharing or integration through a set of rules in a consistent process [HL7]. The HL7 EHR system Functional model provides a reference list of functions described from the user perspective, which may be present in an EHR System (EHRS) to illustrate the granular aspects of functions. The function list designed to enable consistent system functionality. EHRS model neither endorses the technology nor includes the EHR data content but enables information exchange to support the population of “clinical documents, event summaries, minimum data sets and claim attachments”. The EHRS Functional Model is composed of a functional outline (direct care, supportive, and information infrastructure) and a functional profile. There are six rationales for including the function in the EHRS Functional Model, which are: support delivery of effective healthcare, improve patient safety, facilitate
management of chronic conditions, improve efficiency, facilitate self-health management, and ensure privacy and confidentiality. In terms of information infrastructure, EHRS includes seven sections as follows: EHR security; EHR information and records management; unique identity; registry and directory services; support for health informatics and terminology standards; interoperability; manage business rules and workflow [DFH04, MTM11, Spo07].

The Personally Controlled EHR (PCEHR) in Australia has adopted solutions to establish an IT infrastructure for sharing health information. NEHTA applied this to GP vendor systems implemented a standard profile to deliver a PCEHR system across different locations with the application of IHE XDS [Aus12, Neh]. The architectural model is based on a central registry that contains metadata of published documents and an address pertaining to that specific document to facilitate the discovery of documents. The architecture may include several distributed document repositories that enable the document retrieval procedure [NR10]. The healthcare enterprises cooperate for clinical document sharing is called “clinical affinity domain” [DLAE07].

In the field of EHR, Cross-Enterprise Document Sharing (XDS) specification, developed by Integrating the Healthcare Enterprise (IHE) addresses the needs for registration, distribution and access to patient’s clinical information across healthcare enterprises under a document sharing governance structure agreed by all parties involved. XDS provides an integration profile including a registry for querying patient records and methods for retrieving the documents. It employs structured EHR standards such as Continuity of Care Record (CCR) and Clinical Data Architecture (CDA) to facilitate data exchange. This has universally regarded as a method to realise interoperable EHRs, although limited to the exchange of whole documents without being able to analyse their contents. In order to facilitate the application of XDS specification towards the use of ISO 13606, it enables an EHR_EXTRACT to be stored within an XDS repository. The EHR_Extract is used to represent health record information extracted from an EHRS provider and communicated to an EHRS recipient, which contains attributes to identify the subject of care, healthcare provider, the subject’s EHR identifier and the agent responsible for creating it.

IHE solely sets up the foundation for EHR interoperability amongst care domains within single/multiple healthcare enterprises and addresses privacy and security controls through risk assessment and management. Privacy and security are
enabled and enforced at different levels of depth as follows: policy, physical environment, procedures, and organisational, departmental, functional and information technology. It enables three distinct document sharing models, which are: direct push, centralised discovery and retrieve and federated discovery and retrieve. IHE recognises audit trail specifically centralised structure as the primary method of accountability enforcement in the healthcare environment. IHE profile is Audit Trail and Node Authentication (ATANA) that provides security and privacy controls, which are: audit log, identification and authentication, data access control, secrecy, data integrity, non-repudiation, patient privacy and availability controls. The security audit logs, network authentication and encryption for all communications of patient data among trusted systems are established. Furthermore, systems can be connected to each other, once the common policies are mandated, in order to provide a chain of trust in XDS [SSB+12, RCO12].

The CEN/ISO 13606 Electronic Health Communication (EHRCOM) [CEN] is a European norm from the European Committee for standardisation (CEN/TC251) being designed to achieve semantic interoperability in the EHR communication. The overall goal is to define “rigorous and stable information architecture for communicating part or all of the EHR” among EHR systems and centralised EHR data repository. It can be harmonised with IHE XDS, and consequently XDS can store and share 13606 EHR Extract data. Nonetheless, it specifies neither the internal architecture of an EHR system nor the way that data is stored. It also follows a Dual Model architecture that defines an explicit separation between information and knowledge through reference model and archetypes. The openEHR consortium provides the architecture to support distributed, patient-centered, life-long and shared healthcare records. Archetype Definition Language (ADL) is a formal language developed by openEHR for expressing archetypes that adopted by CEN/ISO 13606 [MMM+12]. HL7 can be considered to be the foundation of integrated healthcare environments with CEN/ISO 13606 standard [Beg07].

The ISO 22600: 2014 standard [22614] “defines principles and specifies services needed for managing privileges and access control” to data and functions. This standard is offered in three parts, which are: overview and policy management, formal models and implementations. The focus of this standard is on “communication and use of healthcare data in distributed policy domains include healthcare data sharing”. The ISO 22600-1 is intended to support their technical implementation and also proposes a template with XML for the policy agreement. The structure
includes: domain, policy, roles, directory, authentication and process elements. The rules for the elements are stored in a repository as a part of structure and security and privacy policy domains are distinguished by their policies. The policy ideally should be harmonised and security standards defined at CEN and ISO ought to be the primary tools for achieving this. Otherwise, for each role/ information/ action/ purpose, a set of policies has to be defined with security levels including rules and equivalences between them. This standard is a generic construction of the system including domains and authentication without functional security construction or protocol. It uses cryptography to support digital signatures over a set of assigned attributes. There is a policy ID attribute as references to policies in granularity level and system hierarchy. As stated, any Attribute Authority (AA) can be defined, however it has not specified the implementation.

British Medical Association (BMA) mentioned role-based systems to support restricted access control lists, and in particular containing a single named clinician. Moreover, clinicians must ensure that all record accesses are correctly attributable, marked with the subject’s name, date and time. They set out a security policy based on the rules defined by BMA as the following principles: access control lists, record opening, control, consent and notification, persistence, attribution, information flow, aggregation control, trusted computing-based [And96]. As stated in [And08] central access control policies are appropriate to data-oriented protection systems. Therefore, access control lists are not appropriate when the number of users is large and changing continuously, or where users want to delegate their authority to other users in order to perform specified functions.

Although several Health IT standards have mandated for data sharing, there are still non-standardised communication architectures and models, which have caused semantic divergences. Since healthcare environment is a broad domain with different sub-domains, and a huge number of users, it needs to provide security, interoperability and scalability to share and access EHR data. Accordingly, healthcare providers are responsible for protecting their data to ensure proper access controls are in place. In this chapter, we explore existing studies and introduce a security framework, where the EHR system is managed by multiple authorities, along with a set of protocols for the implementation of the framework. In accordance with the recent development of data sharing, we notice that Attribute-Based Encryption (ABE) can be well fitted into our scenario thanks to its excellent structure which suits well to our framework.
3.1. Introduction

The notion of Attribute-Based Encryption (ABE) was introduced by Sahai and Waters as a solution to enable fine-grained access control for encrypted data [SW05]. The beauty of ABE is that the fine-grained access control is achieved through some cryptographic techniques rather than traditional access control mechanisms. [PTMW10] demonstrated that the ABE system is an efficient solution for securely managing data in large distributed and loosely-coupled systems with the HIPAA compliant distributed file system. Moreover, Ciphertext Policy ABE (CP-ABE) was proposed by [BSW07] to provide complex access control on encrypted data and to keep data confidential even if the storage server is untrusted. This is achieved by embedding the policy in the ciphertexts directly. The main drawback of the ABE system is the basic requirement that needs to have a central authority. Subsequently, [LW11] addressed this issue by proposing a decentralised ABE, which allows multiple authorities to share the same set of attribute policies. This work has enabled new emerging applications of, the large-scale distributed systems, such as cloud computing.

In this work, we enhance this direction of research by proposing a secure and privacy preserved EHR system framework to control the access to EHR data. To illustrate our idea, we incorporate scheme due to [LW11] into our proposed framework to enable such a system. Specifically, we adopt the multi-authority system to our framework to guarantee its practicality. Our EHR system ensures security and scalability features in a distributed environment such as cloud computing. We present the details of EHR data access control using attribute-based cryptography and concerning the secure communication channel between EHR system users and multiple authorities.

3.1.2 Related Work

In the following, we present a review of related work on security and privacy in EHR systems. To start, we highlight the following properties which have been proposed in some of the review studies as follows.

- Flexibility: The data access policies and access structure should be flexible to provide efficient EHR data access, especially in emergencies.

- Scalability: The EHR system should be scalable to provide accessibility for users from public domain/cloud other than the private domain/cloud. The
3.1. Introduction

EHR system scalability could be in terms of key management, storage, access structure, computation and communication.

- Confidentiality: Unauthorised users should be prevented from accessing or decrypting EHR data by proper security implementation, including access control and cryptographic techniques.

- Sharing: The EHR system model should be designed in a way that can share any part of EHR with proper authorisation.

[BHMZ13] proposed an Electronic Transfer of Prescription (ETP) based on National eHealth Transition Authority (NEHTA) with Unified Markup Language (UML). The framework is based on using RM-ODP standards to provide guidelines and support electronic Health systems at Australia’s national level. Their ETP system architecture was proposed for Public Key Infrastructure (PKI) model with credentialing and enrollment functions. Moreover, their interoperable framework facilitates information sharing and NEHTA outcomes’ consistency.

[ARR12] proposed a cloud-based EHR system, which consists of the cloud-based data storage and computing resources, health providers (users), and Attribute Authority (AA). Healthcare providers obtain their private keys from AA to login onto the system with their username and password, and then can encrypt and decrypt EHRs locally through installing lightweight software, and generating access policy using the access policy engine. AA generates the public key and master private key with the setup algorithm and generates a secret key associated with user’s attributes when the new healthcare provider joins the system during the key generation algorithm. In this work, one single AA is responsible for key management including generation, distribution, and revocation in the EHR system. They considered a CP-ABE scheme and organised EHR to the labeled hierarchical data structure to provide flexibility, scalability and fine-grained access control.

In [WHZ13], Private Key Generator (PKG) service for key generation computes the private key of the user which is being used to recover encrypted key for encrypting/decrypting the PHR dataset. Whenever a user wants to retrieve the EHR data set from the cloud, first logs into the cloud server and after passing the authentication flow initiated by the Trusted Third Party (TTP), the server can obtain a private key from PKG to recover the encrypted key. Moreover, after decryption, the TTP transmits the symmetric key to Patient Health Record (PHR) cloud service to
perform key verification and decryption of PHR dataset body. EHR trusted server serves as the root PKG for encrypting EHRs and generating private/decryption keys for EHR owners, domain servers and entities.

The work [BLLS11] in the ESPAC framework (patient-centric access control scheme for e-health in cloud) designed an access control structure which electronic healthcare provider works as a trusted party to do the registration process to generate the keys, and trusted authority assigns unique ID to the healthcare provider. [ZLHL11] proposed an EHR security model with Role-Based and Time-Based Access Control model (RBTBAC) with one Trusted Authority (TA) to provide flexibility. TA contains two parts for encrypting EHR data and enforcing predefined access control policies.

Some studies have aimed to implement cryptographic techniques such as ABE schemes to provide security and privacy of EHR data. [XLZ+15] designed a secure cloud-based EHR system with ABE for efficient storing and sharing PHRs where they applied global authority in the EHR system, responsible for key management, generates keys for physicians, and publishes public parameters for cryptographic operations. In this work, the global authority issues private keys to physicians for retrieving PHRs from the cloud server, so it has all information about users’ secret keys. In another study, [XWC+14] proposed a PHR service system to provide the efficient searching, fine-grained access control, and PHR data sharing with anonymous ABE in the hybrid cloud environment. The TA generates a private key for users and public cloud stores the PHR. There is also a private cloud to facilitate users’ secure usage.

The work in [LBL+12] proposed attribute-oriented authentication and transmission schemes for secure and privacy-preserving health information sharing in health social networks (HSNs). The attribute-oriented transmission scheme enables an HSN user to encrypt the health data associated with access policy which was specified with the set of attributes. The network model shows an attribute trust authority (ATA) to assigning the attribute set and keys.

[HSH12] proposed an EHR data sharing framework that combines Identity-Based Encryption (IBE) and ABE to enforce access control policies and scalable access between different clouds. The domain servers are serving as authorities to enforce access policies and reduce the computation overhead. Then, the trusted server generates and distributes keys to domain servers as a root PKG.
The work in [LYZ+13] presented an ABE-based patient-centric, secure and scalable PHR sharing framework with the security mechanisms for cloud-based PHRs in semi-trusted servers. They applied a combination of [CC09] and [YWRL10] MA-ABE schemes from KP-ABE in which data owners are TA of their data to manage the keys and access rights. The work in [YWRL10] proposed the secure, scalable and fine-grained data access control in the cloud with KP-ABE, proxy re-encryption and lazy re-encryption. In this proposed scheme, the data owner can define the flexible access structure for system users. The work in [IRC12] designed an access control with KP-ABE for e-health systems. There is one TA to generate the encryption and decryption keys. This work also follows a single authority implementation to distribute keys to the users. They provide scalable and simplified key management in their scheme.

The work in [YJRZ13] designed an attribute-based access control for multi-authority systems in cloud storage, whereas there is a globally trusted Certificate Authority (CA) to set-up the system, register all the users and AAs, and assign global user identifier and authority identifier for each authority. A seminal study in this area is the work of [YJ12] that proposed scalable data access control for multiple authority cloud storage systems. While they mentioned their scheme does not require any global authority for key management, there is one CA as globally trusted certificate authority. CA sets up the system and accepts the registration of users and AAs in the system. The CA has the power to assign global identities to users and generate global secret/public key pairs. In [WLL12a] PHR CP-ABE platform, one AA administrates the secret keys based on Waters CP-ABE scheme.

The work in [BCHL09] presented a Patient Controlled Encryption (PCE) system to enable patients to share their partial access rights with others. They aimed to guarantee efficient access, easy sharing and efficient searching over records. Here, patients need to verify healthcare provider’s credentials for issuing keys as trusted parties. [NGS10] proposed a privacy-preserving EHR system with ABE infrastructure to share patients’ data among healthcare providers in a flexible and scalable manner. They applied one TA to generate the private key and a public directory to store public values of the system. To add searchability feature, they combined broadcast CP-ABE with a searchable scheme and a secure channel free Public-Key Encryption (PKE) with keyword search. Moreover, they used flexible protocol for keyword search functionality and ABE for flexible policies. They concluded that
attribute-based cryptographic primitives can provide flexible policies to build secure EHR system infrastructures.

In some proposals [ARR12, And08, CLC+12], the healthcare provider can outsource EHR data and patients can specify access policies in agreement with the TA or healthcare provider. TA is an abstract entity, which is formed by all EHR authorities. TA is responsible for key distribution and issuing credentials of a patient’s PHR. TA is considered as the center to build the access control structure of communication among system users and EHR systems [CLC+12]. In [ZLHL11], TAs are responsible for EHR data encryption from EHR providers into ciphertext format and access control which enforces pre-determined access control policies with a remote EHR database to store the encrypted EHR data.

Although all the studies reviewed so far have attempted to provide security of EHR data, they suffer from the fact that one central TA could lead to a large computation overhead through the system expansion. Moreover, users have different attributes associated with their data which cannot be handled with one single TA in a distributed environment. Notably, when there is one single TA who issues all the secret keys then undoubtedly it has the power to access to all EHR data in the system.

In a follow-up study [Cha07], the multi-authority scheme was presented to support different attribute authorities issuing secret keys to the users for the different set of attributes. Chase improved this idea in another work by removing the trusted central authority which could monitor all the users’ attributes and issues all the decryption keys to the system users. The work in [CC09] proposed multi-authority attribute-based encryption that enables realistic deployment of attribute based access control where different attribute authorities issue different set of attributes. In another study, a decentralised CP-ABE scheme [LW11] was proposed to remove the global coordination of the authorities except the initial setup for common reference parameters and some limitations in access policies of Chase’s scheme. Implementation of decentralised CP-ABE can reduce key distribution and attribute management overhead on one single TA of the proposed EHR system models.

We provide a summary of previous works and our work in Table 3.1. We believe that our framework moves one step forward by introducing CP-ABE with multi-authority. In comparison to other multi-authority systems, we allow authorities to share the same policy base and issue private keys independently. In Table 3.1, we present a comparison of our framework and other proposed EHR systems. The
Table 3.1: Properties of the related work and our framework.

<table>
<thead>
<tr>
<th>Schemes</th>
<th>Flexibility</th>
<th>Scalability</th>
<th>Confidentiality</th>
<th>Sharing</th>
<th>User-based</th>
<th>Multi/Single TA</th>
</tr>
</thead>
<tbody>
<tr>
<td>[BHMZ13]</td>
<td>x</td>
<td>x</td>
<td>v</td>
<td>v</td>
<td>x</td>
<td>Central TA</td>
</tr>
<tr>
<td>[XWC+14]</td>
<td>x</td>
<td>x</td>
<td>v</td>
<td>v</td>
<td>x</td>
<td>Single TA</td>
</tr>
<tr>
<td>[ARR12]</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>x</td>
<td>v</td>
<td>One attribute authority</td>
</tr>
<tr>
<td>[HFLST1]</td>
<td>x</td>
<td>v</td>
<td>v</td>
<td>x</td>
<td>v</td>
<td>Single TA</td>
</tr>
<tr>
<td>[XI2+15]</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>Trusted Server</td>
</tr>
<tr>
<td>[LYZ+13]</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>Multi-TA with KP-ABE</td>
</tr>
<tr>
<td>[YLZ+10]</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>Single TA</td>
</tr>
<tr>
<td>[IRC12]</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>Single TA</td>
</tr>
<tr>
<td>[YJRZ13]</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>Single TA</td>
</tr>
<tr>
<td>[ZLHL11]</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>Single TA</td>
</tr>
<tr>
<td>[WLL12a]</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>Single TA</td>
</tr>
<tr>
<td>[LBL+12]</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>Single TA</td>
</tr>
<tr>
<td>[WHZ13]</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>Single TA</td>
</tr>
<tr>
<td>Our Work</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>Multi-Authority</td>
</tr>
</tbody>
</table>

3.2. Model

We design our model, by making it be as close to Electronic Transfer of Prescription (ETP) as possible. As stated in e-Government Strategy by Australian Government [AGI06], in addition to protecting security and privacy as the users’ requirements, it is preferred to reform the poorly designed and redundant processes, and reduce the duplication by standardising and combining similar processes across agencies. Moreover, [BHMZ13] stated that the ETP system model could be quite complex to demonstrate if the same provider participates in other communities. The ETP architecture can be tailored to the EHR system needs resulting in different solution aspects with different requirements such as security and privacy [BCHL09]. The merit of our approach is about its feature of distributed management, which prevents “single points of failure”. That is, if there is a failure in a TA, other TAs can help and make sure the system functions as normal. The accountability can be secured by the separate private key held by TAs. Any action from a TA is associated with its private key, which ensures the responsibility and accountability of the corresponding TA. Again, this feature has been embedded in the original CP-IBE scheme.

The remainder of this chapter organised as follows. Section 3.2 presents a model, where we describe our framework. Section 3.3 defines the attribute encryption system for our framework. Section 3.4 presents the detailed security protocols. Section 3.5 describes a practical application scenario to further illustrate our framework. Section 3.6 is the conclusion.
ETP system could be improved in a consistent, flexible, scalable and interoperable framework with our solution.

Figure 3.1: System model.

We propose our EHR system model with the decentralised CP-ABE scenario introduced by [LW11]. The model is illustrated in Figure 3.1. The main design goal of our system is to enable different EHR system users having access to EHR data conveniently in distributed environments, such as cloud and enhance privacy and security. The principal users are expected to be authorised healthcare providers, the patient or subject of care who have access to certain functions to their EHR. The healthcare provider receives appropriate decision support to enable effective electronic communication between providers, and between the provider and patient or caregiver.

Whereas there is no standard architecture for EHR systems, we provide a basic model for hospital-based EHR system, including physician, patient, etc. There are different types of data: clinical data, medical and nursing diagnoses, laboratory test results, etc. The EHR database is located in cloud-storage in encrypted form. The system users outsource and retrieve EHR data from EHR cloud-storage. There are multi-authorities who share an attribute policy base. Patient data are encrypted with proper attributes by any of authorities. The authorities, who manage the system, can reside at different locations according to the need. A user can access patient records, according to his/her private encryption key, associated with proper
attributes.

The ETP is a solution specification developed to facilitate interoperability concerned with transferring electronic clinical documents between prescribers and dispensers. The participants in ETP are: subject, prescriber/organization, dispenser/organization, Prescription Exchange Service (PES), subject agent, medications supply manager. This architecture includes issuing authority, registration authority, policy authority, and governance authority for credentialing identities in ETP. The identity system applied is based on the PKI model where enrollment and credentialing are separate functions. On the other hand, as mentioned by [And08] public key certificates are considered to be “crypto” rather than “access control” where their implications for access control policies and architectures are not thought through. In our system, we enable a set of access control policy through different authorities to enable heterogeneous access levels. In this set of policy, we can partition users and resources into domains with distinct administrators, and trust can be inherited between domains (Figure 3.2).

![Figure 3.2: Trust Relationship Diagram.](image)

Our system depicts all the participants in a sound comprehensive and generic communication including healthcare provider (hospital or individual providers) and subject (patient). The individual providers can be doctors and physicians while not any third party such as supply manager is involved in the system, however, it can be registered by authorities. The authorities are set-up to perform the enrollment. There are not any credentialing functions owing to replacing certificate authorities
3.2. Model

by attribute authorities. The access policy is embedded in the attributes which removes the policy authority or any policy management party. We improve privacy by removing the direct access to patient records by the dispenser, supply manager and prescriber. Any other user than healthcare provider and patient itself needs to be registered by multi-authorities to gain access to patient data (Figure 3.3).

![Data Flow Diagram](image)

Figure 3.3: Data Flow Diagram.

Our model is based on decentralised CP-ABE, which is not based on PKI. The benefit of multi-authority is to enable distributed systems and increase the scalability feature by removing the need to CA. Our system provides a rich access control approach to handling complex EHR systems. Our aim is to provide an alternative approach to handling the EHR systems in distributed systems. Avoiding centralised EHR is one of the goals. Our system provides a novel approach for decentralised design. Since it is not based on PKI, our system provides more flexibility to handle access control of EHR. PKI is only applied as part of SSL, which solely provides an authenticated secure channel for data flows (Figure 3.3).

The trust model can be centralised, distributed or federated. Our system model
offers distributed trust model with providing Multi-Attribute Authorities. In accordance with the centralised model, any single point of failure makes a major bottleneck at the central trust, and consequently the system cannot perform. We resolve this issue with Multi-Attribute Authorities implementation. As shown in Figure 3.3, there is a peer-to-peer trust relationship in a distributed model. The direct communication, highlighted by arrows, occurs only between trusted parties who are patient, physician, healthcare provider, patient record server and trusted authorities. The cloud server is an honest-but-curious party, and only provides the required data storage.

### 3.3 ATTRIBUTE-BASED SECURITY FRAMEWORK WITH MULTIPLE TRUST AUTHORITIES

Attribute-Based Encryption (ABE) firstly was introduced by [SW05] where they constructed an Identity-Based Encryption (IBE) of a message under attributes to create a fuzzy identity. There are two forms of ABE presented by [GPSW06, BSW07] namely Ciphertext-Policy Attribute-Based Encryption (CP-ABE) and Key-Policy Attribute-Based Encryption (KP-ABE). In CP-ABE system, keys are associated with the sets of attributes and ciphertexts are associated with the access policies. Then, the user who has the private key which satisfies the policy can decrypt the ciphertext. In the KP-ABE system, private keys are associated with an access structure and the ciphertext is labeled with a set of attributes. Then, when the access structure defined in the private keys matches the attributes labeled with the ciphertext, a user can decrypt the ciphertext. In our system, policies are controlled by the health authority such as the hospital as the healthcare provider. Private/public keys are associated with the sets of attributes (CP-ABE). Our system is based on the original security model of CP-IBE, which illuminates all the potential attacks aiming to compromise the system. These attacks may include: collusion attacks, chosen plaintext attacks (or symmetric security), etc. As demonstrated by Bethencourt [BSW07], CP-ABE is secure against collusion attacks.

**Definition 3.1** (Trusted Authority) Let \(T = \{TA_i\} \) for \(i = 1, \cdots, n\), a set of \(n\) parties, who are fully trusted by all other parties for correctly setting up the system.
and issuing correct private keys to other parties. They share the same set of policy base that consists of a set of attributes. Any trusted authority \( TA_i \in T \) can issue a private decryption key to a user in the system.

**Definition 3.2** (Access Matrix) Let \( A \) be an \( n \times l \) access matrix and maps to its rows to attributes \( \{a_i\} \), for \( n \times l \), be a set of attributes. Any \( a_i \in A \) represents an element used to define the access policy to patient records.

Following the multi-authority Ciphertext-Policy Attribute-Based Encryption system by [LW11], our proposed EHR system is comprised of the following five algorithms:

- **Global-Setup.** This algorithm takes as input the security parameter \( k \) and outputs global parameters \( GP \) for the system.

- **TA-Setup** \((GP) \rightarrow SK, PK\). Each \( TA_i \) runs the algorithm with \( GP \) as input from the Global-Setup phase to produce its own private key \( (SK) \) and public key \( (PK) \).

- **Encrypt** \((M, (A, \rho), GP, \{PK\}) \rightarrow C\). The encryption algorithm takes in a message \( M \), an access matrix \( (A, \rho) \), the set of public keys for relevant \( TAs \), and the global parameters \( GP \). It outputs a ciphertext \( C \).

- **KeyGen** \((GID, GP, i, SK) \rightarrow K_i, GID\). It takes as input a global identity \( GID \) (in our EHR system, it represents a user ID), the global parameters, and an attribute \( i \) belonging to some \( TA_i \) and the private key \( SK \) for this \( TA \). It produces a key \( K_i, GID \) for this attribute and identity pair.

- **Decrypt** \((C, GP, K_i, GID) \rightarrow M\). The decryption algorithm takes in the global parameters \( GP \), the ciphertext \( C \), and a collection of keys corresponding to attribute, identity pairs all with the same fixed identity \( GID \). It outputs either the message \( M \) when the collection of attributes \( i \) satisfies the access matrix corresponding to the ciphertext. Otherwise, decryption fails.

## 3.4 THE PROTOCOLS

We have assumed that our EHR has a set of trusted authorities \( T \), where \( TA_i \in T \) is an abstract entity, which could be a mix of health authorities and hospitals that
control the access control policies in order to provide trusted services. For simplicity, we assume that our EHR system consists of a set of TAs, a set of patients who have full access rights to their own records, a set of physicians who have some or all access rights granted by a TA to their patient records.

**Definition 3.3** (Patient records) Let $R_x$ be a set of patient records wrt patient $x$. Any element $r_i \in R_x$ for $i = 1, \cdots, n_p$ is encrypted with attribute-based encryption using a different or the same $\{PK\}$ depending on the access policy. By $\bar{r}_i$ we denote the encrypted record of $r_i$. Each record is assigned a number or an ID, such as $id_{r_i}$ for record $r_i$.

In Table 3.2, as an example, we assume that all patient records are encrypted with the attribute-based encryption algorithm defined earlier. We assume that each item in a patient record is encrypted with a set of attributes, where they may be different or the same for different records, depending on the access policy. In our scenario, attributes are defined as access rights such as “read”, “write”, “delete”, etc. In light of this, there is no need to know the patient’s records for assigning an access privilege to the user.

<table>
<thead>
<tr>
<th>Patient ID</th>
<th>Record ID</th>
<th>Attributes</th>
<th>EHR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>$id_{r_1}$</td>
<td>1 0 1 ... 1</td>
<td>$\bar{r}_1$</td>
</tr>
<tr>
<td></td>
<td>$id_{r_2}$</td>
<td>1 0 1 ... 0</td>
<td>$\bar{r}_2$</td>
</tr>
<tr>
<td></td>
<td>$id_{r_3}$</td>
<td>0 0 1 ... 0</td>
<td>$\bar{r}_3$</td>
</tr>
<tr>
<td></td>
<td>$id_{r_{np}}$</td>
<td>0 0 1 ... 1</td>
<td>$\bar{r}_{np}$</td>
</tr>
<tr>
<td>Bob</td>
<td>$id_{r_1}$</td>
<td>0 1 0 ... 0</td>
<td>$\bar{r}_1$</td>
</tr>
<tr>
<td></td>
<td>$id_{r_2}$</td>
<td>0 1 0 ... 1</td>
<td>$\bar{r}_2$</td>
</tr>
<tr>
<td></td>
<td>$id_{r_{np}}$</td>
<td>0 1 0 ... 1</td>
<td>$\bar{r}_{np}$</td>
</tr>
</tbody>
</table>

It can be seen from the data in Table 3.2 that all access privileges are associated with unique patient ID, therefore, they cannot be misused. There are record ID associated with EHR data for users, which are unique and enable EHR encryption. ID is unique to a patient. The attributes embedded in the access key of a user are associated with the unique ID, which avoids any potential collusion. Every
user is registered by a TA who assigns proper attributes to the user. “Null” in Table 3.2 represents that patient does not have any specific access right to EHR data. Consequently, the uniqueness of a private key can represent the appropriate access. Accordingly, a user can decrypt the record if his key contains the authorised attributes associated with the correct ID.

In Table 3.3, we provide a summary of participants in our protocols.

<table>
<thead>
<tr>
<th>Party involved</th>
<th>Access rights</th>
</tr>
</thead>
<tbody>
<tr>
<td>TA’s</td>
<td>Grant access rights</td>
</tr>
<tr>
<td>Patient</td>
<td>Read his/her own records only</td>
</tr>
<tr>
<td>Physician/doctor</td>
<td>Read access and/or append new patient records</td>
</tr>
<tr>
<td>Patient record server</td>
<td>Encrypt patient records</td>
</tr>
<tr>
<td>Cloud server</td>
<td>Null</td>
</tr>
</tbody>
</table>

In the following, we provide the precise definition of the entities involved in the system.

**Definition 3.4 (Entities Involved)**

**Patient.** A patient $P$ is the full owner of its own patient record and has the full trust to the set of TAs (Figure 3.3) to correctly manage its record and grant correctly access rights to a physician. Patients have only the “read” right to their own electronic health records.

**Physician.** A physician or medical specialist $D$’s access rights to the patient records are granted by a TA $\in T$. With its access rights to some or all records, $D$ can read these records and append new information to these records but cannot delete any patient record (item). Physicians must show that they are “meaningfully using” certified EHRs by meeting certain objectives.

**Patient Record Server.** Let $S$ be a patient record server, who is responsible to manage patient records. $S$ is a trusted server, who manages and updates patient records according to the access policy. As an important task for $S$, it encrypts patient records with the correct public key (attributes).

Notice that Patient Record Server is not the cloud server, but an authorised server by the health authority. A summary of access arrangement of involved parties is given in Table 3.3.
3.4. THE PROTOCOLS

3.4.1 EHR System Setup

The trusted authorities in $T$ bear the full responsibility to set the EHR system up by firstly using Global-Setup. Upon the completion of Global-Setup, the global parameters $GP$ are used to construct the cryptographic keys of TAs by using TA-Setup. As a result of TA-Setup, each $TA_i$ for $i = 1, \cdots, n$ in $T$ holds a private key $SK$ and a public key $PK$.

3.4.2 Registration

In the registration phase, each user needs to communicate with a trusted authority ($TA_i$) in order to obtain its own private key which we denote as $K_i$, and is generated in terms of the attribute-based policy. Since all trusted authorities share a policy base, any trusted authority can issue a legal key to a user. Suppose that in the initiation phase, an SSL channel is established and all communication flows hereafter are protected. Note that SSL only provides a secure channel for communication as a standard system setup and is not managed by TA. We omit this phase in the following protocols.

**Patient (P) Registration:**

1. $P \rightarrow TA_i: P, REQ_P$.
2. $TA_i \rightarrow P: K_P$.

Here, $REQ_P$ is a request from P for registration to $TA_i$ and $K_P$ is the private key for P to access its own EHR records. $K_P$ is generated by calling the KeyGen algorithm. The KeyGen algorithm takes ID of P, global parameters $GID$ from $TA_i$ global-setup, and attributes of $TA_i$ with private key $K$ for $TA_i$ to output $K_P$ of P.

**Physician/Doctor (D) Registration:**

1. $D \rightarrow TA_i : D, REQ_D(P, id_{r_i}, id_{r_j}, \cdots)$.
2. $TA_i \rightarrow P : K_{D,P}$.

Here, $REQ_D(P, id_{r_i}, id_{r_j}, \cdots)$ is a request from D to obtain private decryption keys for the patient (P)'s record IDs $id_{r_i}, id_{r_j}, \cdots \in R_P$. $K_{D,P}$ is the private key for physician (D) to access health records $id_{r_i}, id_{r_j}, \cdots$. $K_{D,P}$ is generated by calling the KeyGen algorithm. The KeyGen algorithm takes ID of D, global parameters
3.4. THE PROTOCOLS

\[ G_{ID} \] from TA_i global-setup, and attributes of TA_i with private key SK for TA_i to output \( K_{D,P} \) of D.

Record Server (S) Registration:

1. \( S \to TA_i : S,REQ_S \).
2. \( TA_i \to S : PK \).

S receives PK from TA_i, which include all required parameters for encryption of patient records from TA_i. S can encrypt a record on request of a physician or a patient, according to the attribute-based policy by the Encrypt algorithm.

3.4.3 Patient Record Management

Patient Record (Item) Creation:

Creation of a patient record could be done when a patient visits its physician. Optionally, it could be done remotely while the patient consults a physician by a computer network. We assume that the creation of a patient record requires the authorisation of the patient.

1. \( D \to S : E_{PK}(P,r) \).
2. \( S \) encrypts and stores it as \((P, id_{r_i}, \{a_i\}, \bar{r}_i)\).

D generates a record item \( r_i \) for Patient \( P \). \( \{a_i\} \in A \) is a set of attributes, which have been used for the encryption. The patient records stored in the cloud follow the format given in Table 3.2. The encryption is performed with the Encrypt algorithm. The encryption algorithm takes \( r_i \), with record’s ID \( id_{r_i} \), GP of global-setup, and PK of TA_i, then outputs ciphertext \( \bar{r}_i \) and stores as \((P, id_{r_i}, \{a_i\}, \bar{r}_i)\) in cloud-storage.

Reading Patient Record:

We assume that the protocol is executed between a user \( U \) (patient or physician) and S.

1. \( U \to S : P, id_{r_i} \).
2. \( S \to U : \bar{r}_i \).
3.4. **THE PROTOCOLS**

$U$, who can be the patient or a physician, requests patient record $r_i$ to $S$. $S$ finds the encrypted record $\bar{r}_i$ according to $id_{r_i}$ provided by $U$. Upon receiving $\bar{r}_i$, $U$ decrypts it with its private key to $P : K_{U,P}$. The Decrypt algorithm takes ciphertext $\bar{r}_i$, GP of global-setup, and private key $K_{U,P}$, then outputs decrypted record of $r_i$.

**Inserting (Update) to A Patient Record (Item):**

This protocol is executed between a physician $D$ and $S$.

1. $D \rightarrow S : P, id_{r_i}$.
2. $S \rightarrow D : \bar{r}_i$.
3. $D \rightarrow S : E_{PK}(P, r'_i)$.
4. $S$ encrypts and stores it as $(P, id_{r'_i}, \{a_i\}, r'_i)$.

Here, $r'_i$ is the updated record wrt $r_i$. This operation is actually “appending”, where the original content on the record cannot be deleted. Optionally, $S$ can select to use the original $id_{r_i}$, i.e. $id_{r'_i} = id_{r_i}$.

In the setup stage, a proper SSL session is required to provide a secure and authenticated channel, which ensures that all later communication flows are encrypted and all users know that their communication partners are genuine. This process can be easily built into the system, as the SSL can be implemented easily. SSL is merely used for securing the communication channel and plays no role in our access control structure. It can be set up when HTTPS is installed as all other web-based systems. Our protocols only address the access control part. TAs do not require handling SSL connections.

The patient records are all encrypted with proper attributes. Consequently, only the authorised parties can access these records. It is assumed that the cloud server is managed the health authorities who, along with TAs, are trusted by the patients. The cloud server is not authorised to write and update patient records, even though they are trusted to encrypt patient records as an option mentioned earlier. Once a patient record is updated, it must be re-encrypted with the same set of attributes, according to the policy. Once it is encrypted, the cloud server is unable to decrypt.

Global-Setup and TA-Setup are two important algorithms which allow the authorities to run the system. This process is again protected with an SSL session if the communication is required. All late protocols are based on the parameters produced from Global-Setup and TA-Setup.
The patient’s keys are issued by TAs while we have assumed that TAs have obtained these private keys. Hence, they can access all patients’ records. Nevertheless, this is a necessary assumption in order to run the system.

3.5 AN APPLICATION SCENARIO

All patients who receive the treatment in the healthcare system (clinic, hospital, laboratory, etc) must be registered. The authorities, who are in charge of the system act as TAs, are located at different locations. To take one example, there is an authority for each town or suburb. We assume that each patient holds a valid health card (smart card) as, which holds some basic information about the patient, including the private cryptographic keys to access its health record stored in the cloud server. Each patient must be registered with one of the authorities, who issues all information required to use the healthcare system. This information including patient name, address, gender, cryptographic keys, etc. Therefore, patients based on their associated attributes can decrypt their data with their own private key (smart card).

In the registration process, patients are given a unique ID (such as Social Security Number or tax number) and password by a local TA to request for a private key from TA which grants access to their own EHR records (if it is not empty). With its key, a patient has full access rights and is able to modify some of its records, such as allergies or new medications is taking with its own private key, but cannot delete any existing record. Take Alice in Table 3.2 as an example. Her cryptographic key should include the attributes for all her health records of \( (id_{r_1}, \ldots, id_{r_{np}}) \).

Medical staff should be registered and authorised health card (a smart card) is issued to each staff. The smart card contains all his/her patients access keys which can be granted at the registration and updated in the future while a new patient is included and an old patient has left.

A physician should be able to access his/her patients records by defined access policies and with the private key from a TA, read and update the patients’ records in cloud server. Again, no one can delete any record in the system.

If a patient visits a physician in a clinic due to a splinter in thumb, to access the patient history (leg, splinter, etc.), firstly, if not yet registered, the physician should register in an online TA to get its own private key. Physician thereupon can decrypt patient data and access the patient’s medical records from server such as
vaccination, principal diagnosis, insurance, identity, etc. If the finger appears to be infected then the physician needs to assign a blood test or extended reviews. The authorised key of the physician should be able to access the future records of the patient.

Taking Table 3.4 as an example, Alice’s physician, Mike holds a key which contains the attributes to access \((id_{r_2}, id_{r_3})\); therefore, he can access \((r_2, r_3)\) only, but no other records. Mike should hold a table, which contains all his patients’ access information. An example is given in Table 3.4, which shows that he can access Alice’s EHR \((r_2, r_3)\) with key \(ak_{r_2r_3}\) and Bob’s EHR \(r_2\) with key \(bk_{r_2}\).

<table>
<thead>
<tr>
<th>Patient ID</th>
<th>Record ID</th>
<th>Attributes</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>id_{r_2}</td>
<td>1 0 1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>id_{r_3}</td>
<td>0 0 1</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>id_{r_2}</td>
<td>0 1 0</td>
<td>1</td>
</tr>
</tbody>
</table>

### 3.6 CONCLUSION

We have proposed and described an EHR system model which can provide scalability and flexibility of EHR systems in distributed environments while preserving privacy and security of EHR data. We investigated one of the solutions to the security and privacy issues on distributed EHR systems, which avoids using a single trusted authority. With ABE, we can provide fine-grained access control and flexible policies for secure EHR system infrastructures. We plan to develop our EHR system model in more details and with the exchange platforms such as HL7. Although our proposed system meets our design requirements, it is worth noting that some potential issues about the proposed system. Like all systems, which require security protection, the potential computational cost should be considered in future implementation. We assume that TAs and \(S\) have powerful computers, which can handle the additional computation overheads owing to the computation of encryption. Consequently, we proposed that TAs and \(S\) carry out most of the computations while users only require decrypting the corresponding patient’s records. This kind of computation can be easily conducted on a normal PC or even a smart phone. The other issue
we should consider is the coordination in implementation. By virtue of the nature of distributed systems, the policy updates should be managed by the authorised party. We assume that any $S$ can act as the authorised coordinator for the key management.
Chapter 4

Distributed Healthcare Information Sharing with Dynamic Access Control Policy Transformation

Data sharing in Electronic Health Record systems (EHRs) is important for improving the quality of healthcare delivery. Data sharing, however, has raised some security and privacy concerns because healthcare data could be potentially accessible by a variety of users, which could lead to privacy exposure of patients. Without addressing this issue, large-scale adoption and sharing of EHR data are impractical. The traditional solution to the problem is via encryption. Although encryption can be applied to access control, it is not applicable for complex EHR systems that require multiple domains (e.g., public and private clouds) with various access requirements. This chapter addresses the security and privacy issues of EHR data sharing with our novel access control mechanism, which captures the scenario of the hybrid cloud and need of access control policy transformation, to provide secure and privacy-preserving data sharing among different healthcare enterprises. We introduce an access control mechanism with some cryptographic building blocks and present a novel approach for secure EHR data sharing and access control policy transformation in EHR systems for hybrid clouds. We propose a useful data sharing system for healthcare providers to handle various EHR users who have various access privileges in different cloud environments. A systematic study has been conducted on data sharing in EHR systems to provide a solution to the security and privacy issues.
4.1 Introduction

Cloud-based platforms are desirable for delivering electronic health services with ubiquitous network access, scalability and cost saving [SMM14]. Transferring electronic health records (EHRs) to the cloud poses major threats to privacy, data integrity and confidentiality. As a result, several regulations and standards such as the HITECH Act, HIPAA, HL7 CDA, CEN 13606, ISO 22600 EHRcom, IHE XDS and openEHR proposed guidelines and frameworks for sharing and exchanging health information via digital representations of clinical data. However, there are numerous non-standardised communication architectures that have caused semantic divergences. This case reveals that healthcare providers are responsible for protecting their data to ensure that proper access controls are in place [RWS15].

With the growing popularity of cloud computing, EHR data can be stored in the cloud and shared among authorised parties. By using cryptography, secure data sharing can be achieved because encryption can provide a simple form of access control. However, data sharing increases the complexity of key distribution and policy specification of access control. Multi-user settings in the cloud pose challenges to providing an efficient and secure access control mechanism [CLC+12]. Shared data may include patient sensitive and personal information, such as chronic diseases, mental health issues, psychiatric care, sexual behaviour, fertility issues, abortion status and HIV status, which demand privacy preserving implementations. Indeed, encryption as a common practice can provide some basic access control against unauthorised access to private EHR data.

Large EHRs are usually handled by distributed computing systems, such as cloud computing systems. Recently, a large number of papers about EHR access control with Attributed-Based Encryption (ABE) were published. With ABE, one conveniently manages fine-grained access control in the EHR. This has been seen as a promising approach for cloud-based EHR systems. However, in a practical application, EHR data could be stored in multiple clouds due to the need for scalability and privacy. This aspect of EHR systems has not been investigated.

In this chapter, we propose a secure EHR system architecture for secure data sharing, based on several cryptographic building blocks and secret sharing, with Role-Based Access Control (RBAC) to protect patients’ privacy. To better manage the system, we require data to be stored in different types of clouds, i.e. a public cloud and a private cloud. EHRs stored in the private cloud can only be accessed
by the authorised medical professionals, whereas those in the public cloud can be used by medical researchers, pharmaceutical companies, insurance companies, public health agencies, commercial or government agencies, etc. Each cloud requires a RBAC policy that is based on a special type of ABPRE, namely Attribute-Based Proxy Re-encryption. Electronic health record system data sharing is based on the technology of threshold encryption. We consider a practical scenario where an EHR’s data can only be accessed while a threshold number of authorised parties are present. It is usually called threshold secret-sharing. This scenario has been outlined in the literature [Sha79].

We provide an approach for policy transformation for transferring private-cloud policies to the public cloud while encrypted data is transferred. This is necessary when private data must be accessed by different parties. Our hierarchical access structure grants access to authorised users and limits access rights to other users in the public domain. To the best of our knowledge, our approach has not been proposed previously.

The remainder of this chapter is organised as follows. Section 4.2 presents the related work in EHR systems. Section 4.3 introduces our EHR data sharing system scenario, our proposed architecture, the relevant security definitions and our proposed access control policy scheme. Section 4.4 presents an application instance of our proposed access control scheme. Section 4.5 presents the discussion and remarks. Section 4.6 concludes the chapter.

4.2 Related Work

A large body of literature has investigated the issue of data sharing in cloud computing. We summarise them here.

The work of Wu et al. [WAH12] proposed an access control mechanism to support selective sharing of composite EHR data from multiple healthcare providers and preserve patient privacy.

In a study that set out to provide patient privacy and accountability in the health information sharing environment, Ahmed et al. [AAJ14] suggested sharing provenance, which is implementable to the open source CONNECT software to enable eHealth Exchange specifications. Nevertheless, their studies lack the thorough representation of dynamic access control policy solutions.
In another study, Basu et al. [BKL+12] presented Fusion Architecture, an experimental cloud-based platform for securely managing and sharing healthcare information at large scale, however, the access structure to clarify data sharing management and the granting of access by different parties were not presented. Mohan et al. [MBB+09] proposed MedVault as a patient-centric framework for EHR data sharing in which a source-verifiable health record repository evaluates the requests based on the patient’s policy and attributes. Nonetheless, the solution considered neither cloud computing nor policy transformation.

Similarly, Zhang and Liu [ZL10] proposed a security model for sharing and integration of EHR data in the cloud. Encryption and access control were used in the storage server for EHR management with hierarchical and time-bound key management terms. In further studies, [LYZ+13, HMF+13, DYL+14, Hur13] proposed solutions for privacy-preserving data sharing based on ABE or CP-ABE in the cloud to encrypt data and to provide the hierarchical access structure for fine-grained data sharing. They did not provide policy dynamics as in our proposed scheme.

One of the challenges of data sharing is key management. Yu et al. [YWRL10] pointed out data security and access control issues in the EHR sharing within the public domain owing to the heavy computation overhead in key distribution and data management, which occurs in applying fine-grained access control. They used Key-Policy ABE (KP-ABE), Proxy Re-Encryption (PRE), and lazy re-encryption in order to define and enforce access control policies, but secure and dynamic access rights are demanding.

In the same vein, Wei et al. [WLS14] demonstrated a data-sharing system, in which the data holder encrypts data with the public key and then uploads it to the cloud servers, regardless of various access requirements. Furthermore, Chu et al. [CCT+14] proposed a public-key encryption scheme that produces constant-size ciphertexts for efficient delegation of decryption rights in the cloud data sharing in a hierarchical structure. Similarly, in [SF10], a fine-grained access control and searchable public-key encryption technique were applied in an EHR system. A hierarchical access structure was demonstrated to ensure common trust for information sharing.

Calvillo et al. [CRR13] proposed a service-oriented architecture model focused on security and access control in order to empower patients to manage their own health information. Choe and Yoo [CY08] presented a “secure multi-agent architecture” that enables healthcare data access to heterogeneous repositories. A Local
Access Control (LAC) system enables the transformation and administration of access policies by using XML, RBAC and selective encryption. In addition, Chen et al. [CCW10] developed a fine-grained and adaptable access control for healthcare systems through a structured access control rules in XML. In a further study, Duftschmid et al. [DRK +13] undertook an EHR-ARCHE project in order to address the needs to patient’s shared EHR during a treatment process through EHR ISO/EN 13606 archetypes into an IHE XDS environment. Although the aforementioned studies aimed at integrating different hospital policies, the possible security exposures and conflicts were not investigated.

4.3 Proposed Architecture

4.3.1 Data Sharing Scenario

We are interested in a scenario where patients’ data are stored in a private cloud or a public cloud, depending on the access requirements. The data stored in the private cloud can be shared by physicians, but only if a threshold number of authorised parties, e.g. physicians, are present. This feature enabled us to handle the patients’ records that are subject to strict privacy control.

This data-sharing application is particularly designed to provide secure interaction among healthcare parties in large geographical areas and scalable systems. We consider an application of large-scale EHR systems in which the treatment process involves concurrent or sequential treatments of different healthcare givers. This application could be used for continuity of care of patients who need regular check-ups and have emergency episodes in chronic conditions. We illustrate a real healthcare scenario as follows:

When a patient visits a general practitioner in a rural clinic to do a diabetes checkup, he might need to visit a central hospital for major blood tests or require rare medicine from a pharmacy. In an acute episode, he visits the Emergency Department (ED) to receive initial medical. The ED physician then transfers the patient to the central hospital for major tests and hospitalisation. In the treatment process, the clinician in the central hospital, laboratory, pharmacy and clinics need to share and integrate the patient’s health information including treatments, history, test results, primary care visits and emergency care episodes in the case of sharing experiences resulting in patient treatment. If the patient’s data contain sensitive and private
information, two authorised physicians need to be present in order to view or update
the record.

In our solution, we try to facilitate communication and data sharing among the
private domain entities, including hospitals, private clinics, emergency departments
and home healthcare agencies. Meanwhile, we divide the EHR system domain into
two levels, which are respectively handled with two types of cloud: a private cloud
and a public cloud. The private cloud handles the EHR data that are directly used
by medical participants, such as doctors in hospitals and clinics, while the public
cloud handles the EHR data that have been set into a lower level and can be accessed
by external parties (public domain), such as medical researchers and governmental
health authorities.

4.3.2 Our Proposed Architecture

Our EHR system architecture is based on the scenario presented in the previous
subsection with the following requirements:

- Secure EHR data sharing. The protection is required to provide private access
to EHR.
- Privacy enhancement. Additional protection is provided to the private patient
  records by prohibiting their access by a single party. In other words, only a
  threshold number of authorised parties are allowed to access a private record.
- Policy transformation. Flexibility of handling EHR policies is possible using
  our policy transformation approach.
- Fine-grained role-based access control. Fine-grained role-based access control
  is provided by using ABE-based technology.

As seen from Figure 4.1, the proposed framework utilises two types of clouds:
a private cloud and a public cloud. The server in the private cloud \(S_{pri}\) contains
private EHRs, which can only be accessed via the private domain by doctors, spe-
cialists, patients, etc. The server in the public cloud \(S_{pub}\) contains the information
that might be used by other parties or public domain such as researchers, educators,
insurance companies, government, etc. The access control mechanisms are differ-
ent in these clouds due to different security requirements. In the private cloud, we
utilise the mechanism of RBAC and secret sharing to protect the patients’ privacy.
In the public cloud, we adopt the scheme of RBAC, but it requires a different policy. Therefore, we introduce an approach for policy transformation (TR). As the policy for the private ($Pol_p$) is transferred to that for the public cloud ($Pol_c$), the encrypted data need to be transferred to the public cloud, too, which requires the attribute-based proxy re-encryption technology [LCLS09]. There is a trusted system server ($S$), who sets up the entire system. $S$ (alternatively, $S_{priv}$) can also be the proxy who acts as the party, which converts the encrypted data from the private cloud to the new encrypted data stored in the public cloud. This transformation can be conducted with the scheme of proxy re-encryption. One of the challenges to policy transformation and data conversion lies in the difficulty of the task of converting the policy and shared data to the policy with unshared data, where we assume that secret sharing in the public cloud is not required. In light of the above, our framework is novel and shows a new way to provide security and privacy to EHRs. Although several previous works also suggested using RBAC and secret sharing, they adopted different ways of handling security and privacy in EHRs.

![Figure 4.1: The framework with policy transformation.](image)

### 4.3.3 Definitions

In this section, we start by defining the basic components for our system and then define our policy model along with the policy transformation and examples.

**Definition 4.1 (Subject)** Let $S$ be a set of subjects which are active. Let $s_i \in S$ be a subject variable, which is a user.

A subject is an active user and plays a pivotal role in our system. For example, it could be a doctor who accesses a patient record according to the access control policy.
Definition 4.2 (Role) Let \( R \) be a set of roles and \( \text{rol} \in R \) be a role variable, which can be assigned to a user.

A role could be doctor, nurse, dentist, etc. Every user in the system is assigned a role, which is subject to a permission defined as a tuple \((r, w, d, x)\) meaning read, write, delete and execute respectively. If permissions are shared, we denote it as \(sh\tau_n(r, w, d, x)\), stating that requiring \( \tau \) of \( n \) users to act together to have the permissions. When a role is assigned to a user, then the user is granted with the corresponding permission.

Definition 4.3 (Object) Let \( O \) be a set of objects, where \( o_i \in O \) for \( O = \{o_i\} \), \( i = 1, \cdots, n \), a set of \( n \) EHR data.

Objects or EHR data in the access control policy are targets, which are passive. Their privacy and security are achieved through the access control mechanism. The key feature of our approach is “policy dynamics”, which captures the variation of policies. In order to express the policy dynamics, we require a policy to be also a function of time and domain. By policy dynamic, we mean that policy changes from one domain to another without compromising security and privacy. Here, domain refers to the environment where the policy is applied to. We define our syntax by the role-based model.

Definition 4.4 (Policy) Access control policy \( \text{Pol} \) defines who gets access to what. Policy is a tuple \( \text{Pol} \subseteq R \times O \times P \times T \times D \times PU \), where \( R, O, P, T, D, PU \) are sets of roles, objects, permissions, time space, domains, and purposes respectively. As an instance of \( \text{Pol} \), \( \text{pol} := (\text{rol}, \text{o}, \text{p}, \text{t}, \text{d}, \text{pu}) \) where \( \text{o} \in O \) is a set of EHR data associated with a patient, \( \text{p} \in P \) is the permission, \( \text{t} \in T \) is the time (start/end times), \( \text{d} \in D \) is the domain and \( \text{pu} \in PU \) is the purpose of access.

Our policy setting is role-based, where each role is pre-assigned to the attributes defined in \( \text{pol} \). If a user is assigned to a role, it will have all the rights specified in \( \text{pol} \).

Definition 4.5 (Policy Transfer) Policy Transfer \( \text{TR} \subseteq \text{Pol} \times \text{Pol} \). As an instance of \( \text{TR} \), \( \text{tr} := (\text{pol}_p, \text{pol}_c) \), where \( \text{pol}_p \) denotes the previous policy and \( \text{pol}_c \) denotes the current policy. As an instance, \( \text{tr} \) transfers \( \text{pol}_p \) to \( \text{pol}_c \), denoted by \( tr : \text{pol}_p \rightarrow \text{pol}_c \).
4.3. Proposed Architecture

For example, let $tr = (pol_p : pol_c) \subset TR$, for

$$pol_p := (\text{physician}, *, \text{update}, *, \text{private\_cloud}, *)$$

and

$$pol_c := (\text{researcher}, \text{MedicalHistory}, \text{read}, 20150101-20160101, \text{public\_cloud}, \text{research}),$$

where $*$ denotes an arbitrary value of the corresponding attribute. The previous policy $pol_p$ is transferred into the current policy $pol_c$ by the transformation operator $tr$. $tr$ plays an important role in the transformation. In the following sub-section, we will explain how it is realised using cryptographic building blocks.

4.3.4 Access Control and Transformation of EHR Policy

Our approach is based on some existing building blocks, which allow us to concretely conduct policy transformations. The fine-grained RBAC scheme is based on Ciphertext-Policy Attributed-based Encryption (CP-ABE), which can be used to define the detailed access structures to EHR data meeting the RBAC policy. Owing to the requirement of policy transformation, we consider one special type of CP-ABE, which is called Attribute-Based Proxy Re-Encryption [LCLS09]. The shared EHR data are based on the threshold encryption [BBH06], which is based on the secret-sharing scheme proposed by Shamir [Sha79].

To demonstrate our access control, Table 4.1 shows a data access policy in the private cloud, which is also used as an example of the policy before transformation. We see that it is transferred into a public policy (Table 4.1), with a policy transformation. The permissions in the tables are sets of attributes for ABE we adopt. $o_i$ denotes an EHR file; roles are sets as Physician, Physician2 and Nurse. Physician2 is the role which requires shared access to the EHR $o_2$, where $sh_2^2$ denotes that at least 2 out of 5 physicians are required to obtain access to $o_2$; the permissions are defined as read ($r$), write ($w$), execute ($x$) and delete ($d$). Here, $\bar{o}_i$ denotes an encrypted file.

After the policy transformation, the transferred policy given in Table 4.2 has a new set of attributes (permissions), wrt new roles: Researcher, Auditor and Insurance Company respectively, which are permitted to read the corresponding EHR files with a new access key. Here, $\hat{o}$ denotes re-encrypted record file $o$ (from $\bar{o}$).

To realise our idea, we utilise several cryptographic building blocks. For the convenience of presentation, we define these building blocks in the following definitions:
Table 4.1: The private cloud access table.

<table>
<thead>
<tr>
<th>Role</th>
<th>Key</th>
<th>Permission</th>
<th>EHR ID</th>
<th>EHR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physician</td>
<td>$K_1$</td>
<td>$r$</td>
<td>$w$</td>
<td>$x$</td>
</tr>
<tr>
<td>Physician2</td>
<td>$K_2$</td>
<td>$sh^2$</td>
<td>$r$</td>
<td>$w$</td>
</tr>
<tr>
<td>Nurse</td>
<td>$K_3$</td>
<td>$r$</td>
<td>$\neg w$</td>
<td>$\neg x$</td>
</tr>
</tbody>
</table>

Table 4.2: The public cloud access table.

<table>
<thead>
<tr>
<th>Role</th>
<th>Key</th>
<th>Permission</th>
<th>EHR ID</th>
<th>EHR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Researcher</td>
<td>$K'_1$</td>
<td>$r$</td>
<td>$\neg w$</td>
<td>$\neg x$</td>
</tr>
<tr>
<td>Auditor</td>
<td>$K'_2$</td>
<td>$r$</td>
<td>$\neg w$</td>
<td>$\neg x$</td>
</tr>
<tr>
<td>Insurance Company</td>
<td>$K'_3$</td>
<td>$r$</td>
<td>$\neg w$</td>
<td>$\neg x$</td>
</tr>
</tbody>
</table>

**Definition 4.6** A threshold encryption (TE) is a function $TE(K, M)$, which takes as input the encryption key $K$ and the message $M$ and outputs a ciphertext $C_1$. The decryption of $C_1$ is a function $TD(s_1, \ldots, s_t)$, where secret shares $s_1, \ldots, s_t \in \{s_i\}_{i=1}^n$ and $t (\leq n)$ is the threshold.

$TE$ is used as threshold secret sharing, i.e. only $t$ of $n$ users can collaboratively access the encrypted file in order to enhance privacy in the private cloud.

**Definition 4.7** (Encryption) Attribute-Based Proxy Re-Encryption (ABPRE) is represented by a function $ABPRE(key_1, M)$, with the input of attribute-based encryption key $key_1$ and message $M$ and outputs a ciphertext $C_2 = ABPRE(key_1, M)$.

We require an ABPRE scheme for encryption of patient records. ABPRE presents a nice way to implement the role-based access control while the most important feature of ABPRE is to allow re-encryption by a third party (proxy) without using the original private key. With this feature, we can build the policy transformation scheme. Definition 8 defines the re-encryption based on ABPRE.

**Definition 4.8** (Re-encryption) ABPRE is a function $ABPRE(rekey, C_2)$, where on input of a rekey and ciphertext $C_2$, it outputs a new ciphertext $C_3 = ABPRE(key_2, C_2)$, where rekey is associated with key1 and key2.
The encryption and re-encryption algorithms are described in the following algorithms (Algorithm 1 and Algorithm 2).

**Algorithm 1:** Encryption

**Data:** C2 = 0, C3 = 0;
**while** Policy = true **do**
  **read** K, key1, M;
  **if** Condition = true **then**
    C2 = TE(K,M);
    C3 = ABPRE(key1,C2);
    return C3;
  **else**
  Continue;
end

**Algorithm 2:** Proxy-re-encrypt

**Data:** C2 = 0, C3 = 0;
**while** not at end of the EHRs **do**
  **read** current EHR;
  **read** key2;
  **read** C;
  **if** C ≠ 0 **then**
    C" = ABPRE(key2, C);
    return C";
  **else**
  Continue;
end

Recall that policies are associated with encryptions, i.e. to access an encrypted EHR, a party needs to have a key, which corresponds to a set of authorised attributes (policy). To convert a current policy to a new one, the trusted authority applies the proxy re-encrypt algorithm to form a new ciphertext that is encrypted using a new key corresponding to a new set of attributes, according to the access policy of the new environment. In our EHR system, the current policy is referred to as the policy before transformation on the private cloud. However, the new policy is referred to
the same EHR downgraded to the public cloud, i.e. encrypted with a new key which corresponds with a new set of attributes.

4.4 An Instance of Application

Let us assume that the following participants are involved.

- The TTP or Server ($S$). It is an honest-but-curious participant who is responsible for setting the system up, including key generation, registration, maintenance, etc. Note that the cloud service provider should not be able to access plain EHR files in our system, but it is reliable to provide the storage service.

- Server ($S_{priv}$): An honest-but-curious participant that manages private cloud.

- Server ($S_{pub}$): An honest-but-curious participant that manages public cloud.

- Doctor ($D$). Doctor is a role, who treats its patients. A doctor could be a physician who resides in a hospital, clinic, mobile medical service, etc.

- Patient ($P$). Patient is a role, who is treated by a doctor, and the patient’s medical records are stored in the cloud(s).

- Researcher ($R$). Researcher is a role, who acts as a user and has access to the designated patient’s records stored in the public cloud.

- Private cloud ($PriCloud$). Private cloud is a domain, which provides private storage for the EHR system.

- Public cloud ($PubCloud$). Public cloud is a domain, which provides storage for the EHR system and can be used by “public” users such as researchers, statisticians, etc.

The application protocol is implemented as follows. Suppose that all communication flows are protected by a normal SSL channel, so they are secured against eavesdropping.

- System Setup

  The $S$ is trusted authorities that bear the full responsibility to set the EHR system up. The $S$ selects a pair of public/private keys ($PK, SK$) that are used
to compute other participants’ keys. There is a set of attributes $a_i \in A$ for $i = 1, ..., n$ as input for the key generation algorithm. An example of attributes is $(r, w, x, d)$ as defined earlier in this chapter.

- **User Setup**

  1. $U \rightarrow S$: $U, \text{REQ}_U(id_i), i \in \alpha$.
  2. $S \rightarrow U$: $SK_U$.

A system user $U$ in the private domain, such as doctor or patient, sends a request to the server to be registered for the first time. Here, $\alpha$ is a set of indices wrt the EHR. $\text{REQ}_U(id_i)$ is a request from user $(U)$ to obtain private decryption keys for records $id_i \in \alpha$. $SK_U$ is the private key for $U$ to access the requested records $o_i$ for $i \in \alpha$ in the private cloud. $SK_U$ is generated by calling the corresponding key generation algorithm with proper attributes and the role assigned to the user.

- **Researcher Setup**

  1. $R \rightarrow S$: $R, \text{REQ}_R(id_i), i \in \gamma$.
  2. $S \rightarrow R$: $SK_R$.

Here, $\gamma$ is a set of indices wrt the EHR. $\text{REQ}_R$ is a request from researcher $R$ in the public domain to obtain private decryption keys for record $\hat{o}_i$. $SK_R$ is the private key for $R$ to access record $\hat{o}_i$ in the public cloud. $SK_R$ is generated by calling the corresponding key generation algorithm and the role assigned to $R$.

- **Reading Patient Record**

  We assume that the protocol is executed between doctor $D$ and $S_{priv}$ in the private cloud $PriCloud$.

  1. $D \rightarrow S_{priv}$: $\text{REQ}_D(id_i)$.
  2. $S_{priv} \rightarrow D$: $\hat{o}_i$.

$D$, who is a doctor, requests patient record $o_i$ wrt $id_i$ to $S_{priv}$ in the private cloud $PriCloud$. $S_{priv}$ found the encrypted record $\hat{o}_i$ wrt $id_i$. Upon receiving
4.4. An Instance of Application

\( o \), \( D \) decrypts it with its private key \( SK_D \): \( o \). The algorithm takes \( o \) and \( SK_D \) as input and returns \( o \) if it can be decrypted, otherwise it returns \( \perp \).

\[
pol_p := (D, o, \{r, x, w, -d\}, *, PriCloud, *)
\]

- **Reading Shared Patient Record**

We assume that the protocol is executed between doctor \( D \) and \( S_{priv} \) in the private cloud \( PriCloud \).

1. \( \{D_j\}_{1}^{t} \rightarrow S_{priv}: \text{REQ}_{D_j}(id_i) \).
2. \( S_{priv} \rightarrow \{D_i\}: sh^t_i(\bar{o}_i) \).
3. \( \{D_j\}_{1}^{t} \): collaboratively compute \( o_i \).

Here, a set of doctors send a request to \( S_{priv} \) to access the shared record \( \bar{o}_i \) wrt \( id_i \). Then, \( S_{priv} \) sends \( sh^t_i(\bar{o}_i) \) to these doctors, where at least \( t \) members of \( \{D_i\}, i = 1, \ldots , n \) can collaboratively compute \( o_i \).

- **Updating Patient Record**

This protocol is executed between a doctor \( D \) and \( S_{priv} \) in the private cloud \( PriCloud \).

1. \( D \rightarrow S_{priv}: o'_i \).
2. \( S_{priv}: \bar{o}'_i \) (store \( \bar{o}'_i \) and update the policy accordingly).

Here, doctor \( D \) stores the updated record \( o'_i \) in the private cloud \( PriCloud \). \( \bar{o}'_i \) is the updated record wrt \( \bar{o}_i \) with attributes \( a_i \). This operation is actually “appending”, where the original content on the record cannot be deleted. Note that we have assumed that a secure and authenticated SSL channel has been set up for conventional security. Therefore, the transmission of \( o'_i \) is secured. We assume that the encryption is conducted by \( S_{priv} \).

- **Policy Transformation**

\[
S_{priv}: \bar{o}_i = ABPRE(key2, \bar{o}_i) \leftrightarrow ABPRE(key1, \bar{o}_i),
\]

\( S_{pub} \) sends a request to private cloud \( S_{priv} \) for patient record of \( o_i \). \( S_{priv} \) (can also be the proxy) looks up for encrypted record \( \bar{o}_i \) wrt \( o_i \) for re-encryption. \( S_{priv} \) takes re-encryption key \( key2 \) and the ciphertext \( \bar{o}_i \) as input and outputs
Therefore, the previous policy $pol_p$ has been transferred into the new policy $pol_c$. That is, $tr : pol_p \rightarrow pol_c$. As an example, let the policy from the private cloud be:

$$pol_p := (\text{Physician}, \hat{o}_1, \{r, w, x, \neg d\}, \ast, PriCloud, \text{history})$$

The new (current) policy after transformation reads:

$$pol_c := (\text{Researcher}, \hat{o}_1, \{r, \neg w, x, \neg d\}, \ast, PubCloud, \text{research})$$

- **Reading A Record by Researcher**

  $R$ holds the key that meets:

  $$pol_c := (\text{Researcher}, \hat{o}_1, \{r, \neg w, \neg x, \neg d\}, \ast, PubCloud, \text{research})$$

  where the key held by $R$ is embedded with the attributes $\{r, \neg w, \neg x, \neg d\}$.

  1. $R \rightarrow S_{pub}$: $REQ_R(id_i)$.

  2. $S_{pub} \rightarrow R$: $\hat{o}_i$.

  Once $R$ received $\hat{o}_i$ from $S_{pub}$, he can decrypt it with his key and read the corresponding patient record.

### 4.5 Discussions and Remarks

We accomplished our objective of a novel role-based access control approach with the flexibility of dynamic policy transformation and threshold sharing. We anticipate that it can be implemented in any hybrid cloud system by healthcare providers. Our purpose is to simulate our solution because our future work is in a multiple domain system, including a clinic and hospital, in which multiple physicians and other medical staff are required to access electronic patient records. We also plan to implement the threshold sharing as an option, which allows the service provider to decide the best practice and privacy for the system. The security properties of the cryptographic building blocks used in our system have been mathematically proven in the literature [Sha79, LCLS09, BBH06].

Although there are inevitable risks to data sharing, including complex policies of EHRs data and the increase in the number of users in general, our solution has
merits of providing scalability, confidentiality and secure data outsourcing. Some of the foreseen drawbacks to the solution might be potential large overheads for data dissemination and data retrieval, and different data formats in different healthcare repositories. These issues should be able to be solved with a proper cloud system and software engineering.

Numerous studies have been undertaken to address the issue of data sharing in cloud computing with fine-grained access control, such as [DYL+14] and [LHL15], whereas they considered trusted clouds or single cloud which do not provide dynamic policy update. Some of the investigated issues in related works include unauthorised accesses from users of multiple roles, computational complexity, threats to highly private and confidential EHR data, etc.

We have employed a threshold scheme as an option to increase patient privacy, but there is no doubt that when the threshold number is 1, it returns to a normal encryption. In the literature, there are some attempts to introduce threshold secret sharing to EHR systems. For example, Yi et al. [YMBW13] employed a threshold encryption scheme to protect the privacy of patient’s data under the multi-party framework where multiple parties cooperate to control EHR data access without compromising privacy and security. In addition, Yu and Hou [YH14] proposed a certificate digital signature scheme based on the Shamir’s threshold scheme and the Schnorr’s digital signature scheme to enhance the trustworthiness. In [EF13], Ermakova and Fabian presented an architecture for sharing EHRs in a cloud with the help of real case study. They designed a multi-provider cloud architecture that features Shamir’s secret sharing to distribute EHRs to various cloud services. However, our threshold approach is built on hybrid cloud and dynamic policy transformation, which demonstrates a nice application for threshold sharing.

We have not tested our proposed solution yet, as it requires substantial resources to support a test. As future work, we will conduct a pilot study to test our system. Consequently, we require a hybrid cloud system (public and private) to host that has the following components:

- The server programs that implement the role-based access control scheme,
- EHR databases for all the patient records and physicians information,
- Cryptographic units to implement encryption and other schemes, i.e. secret-sharing required in our system, and
4.6 Conclusion

- Data communication units to coordinate data flows amongst all participants.

On the client side (patient, physician, and all other users served by the system) requires a client-side program, which communicates with the server programs according to the established protocols. The client-side program should, at least, host the following components:

- An access control unit, which implements the access control mechanism in cooperation with the server programs,

- A cryptographic unit, which implements all cryptographic operations required to clients, i.e. secret-sharing scheme, RBAC, and

- A communication unit, which handles all communication schemes with the servers.

The test of efficiency and reliability of the proposed schemes is the main goal. The testing results will provide a comprehensive overview of the requirement of the system capacity, including computation capacity and communication capacity.

4.6 Conclusion

We presented an access control mechanism for an EHR system with the hybrid cloud structure, which allows us to handle various types of users who possess different access privileges. Our system features dynamic policy transformation based on some useful cryptographic building blocks. Our novel policy transformation approach enables EHR data to be transferred from a private cloud to a public cloud with the corresponding transformation in the access control policy. We proposed an implementation protocol for an application scenario.
Chapter 5

Access Control Policy Combination from Similarity Analysis for Secure Privacy-preserved EHR Systems

In distributed systems, there is often a need to combine the heterogeneous access control policies to offer more comprehensive services to users in the local or national level. A large scale healthcare system is usually distributed in a computer network and might require sophisticated access control policies to protect the system. Therefore, the need for integrating the electronic healthcare systems might be important to provide a comprehensive care for patients while preserving patients’ privacy and data security. However, there are major impediments in healthcare systems concerning not well-defined and flexible access control policy implementations, hindering the progress towards secure integrated systems. In this chapter, we introduce an access control policy combination framework for EHR systems that preserves patients’ privacy and ensures data security. We achieve our goal through an access control mechanism which handles multiple access control policies through a similarity analysis phase. In that phase, we evaluate different XACML policies to decide whether or not a policy combination is applicable. We have provided a case study to show the applicability of our proposed approach based on XACML. Our study results can be applied to the Electronic Health Record (EHR) access control policy, which fosters interoperability and scalability among healthcare providers while preserving patients’ privacy and data security.

5.1 Introduction

An Electronic Health Record (EHR) system is “a repository of information regarding health status of a subject of care”. It combines data from various distributed healthcare sectors [GBL11]. Since the adoption of EHRs by healthcare providers
has emerged, there has been a demand to protect patient privacy and security of medical data through the proper access control policies. Access control policy has a paramount importance in healthcare systems security to protect medical data. It may permit the user to get unauthorised access, exposing all unwanted information and patient privacy. Thus, well-defined access control policies are essential for providing confidentiality and limiting the unauthorised access rights to patient’s data.

The majority of the access control models and policies in EHR systems have adopted ontological solutions or cryptography techniques for the sake of the constraint of patient data access, which cannot provide flexible access rights integration and pervasive services. A patient data might be subject to the management of several data systems, to receive different types of medical services in the interim, for instance. As a consequence, there are different versions of EHR and established access control policies in healthcare systems. On the other hand, access control models may need to be combined for the integrated and up-to-date systems [WAH12, KKTT14] for better access policy management and comprehensive access while preserving patient’s data security and privacy. There exists a gap in the composition of EHR systems’ access control policies, which could result in inadequate delivery of care and users’ privacy exposure.

In this work, we contribute an access control policy combination solution that fosters interoperability and scalability of access control services among different enterprises and preserves patients’ privacy. While there are different access control policies for an EHR established in different entities, we propose a scheme to compute policy similarity of two or more access control policies in EHR before combining policies. Similarity evaluation process plays a crucial role in preserving users’ privacy in distributed environments [LRF13].

We aimed at a more efficient and lightweight approach than previous works to measure policy similarities. Our proposed approach can be utilised to compare different XACML policies and evaluate the combination applicability with low computation complexity. Our access control system framework introduces a new mechanism to enforce access policy combination, which is applicable to across different healthcare providers with different types of access control policies.

We illustrate an EHR system scenario and the implementation of the scheme in XACML. XACML possesses a policy combination scheme, which defines the eligibility of policy combination and rules. Our goal is not redefining how rules should
5.2. Related work

Several efforts have been devoted to similarity analysis, policy combination and analysis of access control policies. In a study conducted by Lin et al. [LRBL07], policy similarity measure is used as a filter for policy similarity analysis and evaluation of policies. They classified attributes as different types of categorical attributes and numeric attributes. In a follow-up work, Lin et al. [LRB+10] proposed a framework called EXAM which consists a policy analyser component in order to perform different analysis queries expressed in XACML. As Lin et al. [LRF+13] stated, they proposed a lightweight ranking approach to locating parties with similar policies. They applied the same method for classifying attributes and computing the similarity score with the proposed case study.

In a recent work, Bertolino et al. [BDK+15] performed access control test prioritisation relying on similarity criteria. They used similarity matrix to compute dissimilarity among different policies and then calculate policy similarities.

With respect to the policy combination, Li et al. [LWR+08] proposed a Policy Combining Language (PCL) to express Policy Combining Algorithms (PCA) in XACML. They applied evaluation of PCAs using finite state automata. In a seminal study in this area [RLB+11], Rao et al. proposed an algebra for fine-grained integration of policies with a framework to use the algebra in XACML policies. In this study, a set of operators used to edit a policy with pre-defined properties. In a study which set out to determine policy integration [MBC06], Mazzoleni et al. proposed an XACML policy integration algorithm to enable collaboration of autonomous subjects sharing their sources. They computed rule similarity based on policy rule similarity types. Then, the policy integration performs based on rule effects. In addition, Don et al. [DMS+11] proposed a privacy policy aggregation in
P3P. They attempted to capture privacy issues in service aggregation with the same concepts.

A large and growing body of literature has investigated access control models and policies in healthcare systems with cryptographic and non-cryptographic techniques. There are many access control mechanisms proposed in EHR systems, such as broker-based access control. There are different types of access control models, including Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC) and Cryptographic Access Control (CAC). RBAC is the most common access control model in healthcare systems [ASLT13], however, ABAC has been a desired option due to its flexibility in the policy descriptions. Notably, among all different access control strategies, XACML is an open standard of access control policy languages, which has been desirable to define flexible access control policies in healthcare systems [RWS15].

In [MA12], a system architecture and protocols to enable patient control over its health information have been proposed. In this study, Public Key Encryption (PKE) used to secure patient data. In the same vein, Chen et al. [CLC+12] proposed an access control scheme using Symmetric Key Encryption (SKE) which supports multi-user access dynamically in cloud computing. Moreover, Zhang et al. [ZLX14] proposed a role-based and time-based access control with spatial and temporal dimensions to improve fine-granularity of time-bound access control.

5.3 EHR system scenario

We consider a case (Fig. 5.1) in which there are different pre-established access control policies in an EHR system associated with different types of treatments. As an instance of policies, there are different types of insurance policies in order to cover patient treatment including general treatment policy and hospital policy. The hospital policy covers the patient treatments when he goes to a hospital, whereas general treatment policy covers him when goes to a medical centre for an ancillary treatment purpose, such as optical treatment. These two policies are set by two different users on the same patients’ data, which can be in the same healthcare provider or different. If a patient visits an optical specialist in a hospital to perform an eye laser surgery, a comprehensive cover for the patient treatment will be desired. Hence, general treatment policy and hospital policy are needed to be combined as “Mix and Match” to offer a packaged policies that can cover both hospital and
general treatment services and preserves patient’s privacy. The advantages of this solution is a high quality and time-saving continuity of care, which undoubtedly ensures patient’s data security and privacy.

As shown in Fig. 5.1, the Access Control Server (ACS) plays the XACML Policy Enforcement Point (PEP) role as the service interface in [OAS09]. There are authorities as Policy Information Point (PIP) that store users’ access rights. The PIP sub-components are repository, policy authoring function, policy testing and analysis function, policy distribution function and Policy Decision Point (PDP)[OAS15]. Access control databases or Policy Administration authority Point (PAP) interacts with PIP in making policies available to access control Policy Combination Service (PCS). The PCS acts as PDP, which interacts with PAP and PIP to compute policy combination based on access control policy combination framework and interacts with ACS for enforcing the combination function (PEP).

5.4 Proposed policy combination system

Our policy combination framework is depicted in Fig. 5.2. It shows that XACML policies in the form of PAP are used by Policy Combining Service (PCS) to check the similarities and compute the Combination Applicability Value (CAP). Policy Similarity Analyser (PSA) computes the similarity score among sets of policies. Our system model enables the policy owner to specify the effects and attributes
in order to compare the policies. We illustrate the essential components which are needed to be analysed in PSA, including policy, policy-sets, rule and rule-sets in XACML.

It is essential to apply combining algorithms in both policies in order to measure combination applicability. This plays a pivotal role in protecting data against unauthorised access and preserving users’ privacy. In our system, a Policy Combination Algorithm (PCA) allows policy combination according to the combination applicability value. It is worth noting that the attribute overriding specification can be placed in the XACML data-flow diagram to prioritise each attribute including subjects, resource, action and environment, leading to a fine-grained policy similarity analysis. The similarity analysis phase measures the similarity of attributes and effects in the given policies based on similarity formulas.

Policies can be combined based on comparing the RCA and PCA in the proposed Policy Combination Service (PCS) (Fig. 5.2). The Rules Combination process consists of the Similarity Analysis that checks the attributes, conditions and effects of the set of rules in the given policies, in order to measure the similarity value.

![Policy Combination Framework in XACML](image_url)

Figure 5.2: Policy Combination Framework in XACML.
5.4. Proposed policy combination system

5.4.1 Definitions

Before introducing the policy combination algorithm, we present the definition of PolicySet, Policy and Rule.

**Definition 5.1 (PolicySet)** Let PolicySet be a set of \( \text{Pols} \subseteq T \times \text{PCA} \times \text{Pol} \) where \( T, \text{PCA}, \) and \( \text{Pol} \) denote Target of PolicySet, Policy-Combining Algorithm and set of Policies, respectively. As an instance of \( \text{Pols} \)

\[
\text{Pols} := \{\{t_1, \ldots, t_n\}, \{D, P, NA, Ud\}, \{\text{pol}_1, \ldots, \text{pol}_m\}\},
\]

where \( \{t_1, \ldots, t_n\} \) denotes a set of attributes of target, \( D \) deny, \( P \) permit, \( NA \) non-applicable, \( Ud \) undecidable, and \( \{\text{pol}_1, \ldots, \text{pol}_m\} \) a set of policies.

**Definition 5.2 (Policy)** Let Policy be a tuple \( \text{Pol} \subseteq T \times \text{RCA} \times \text{R} \) where \( T, \text{PCA}, \text{R} \) are Target, Rule-Combining Algorithm, and Rule, respectively. As an instance of \( \text{Pol} \)

\[
\text{Pol} := \{\{t_1, \ldots, t_n\}, \{D, P, NA, Ud\}, \{r_1, \ldots, r_m\}\},
\]

where \( \{t_1, \ldots, t_n\} \) denotes a set of attributes of target, \( D \) deny, \( P \) permit, \( NA \) non-applicable, \( Ud \) undecidable, and \( \{r_1, \ldots, r_m\} \) a set of rules.

**Definition 5.3 (Rule)** Let Rule be a set of \( \text{R} = \{T, C, E\} \) where \( T, C, E \) are Target, Condition and Effect. An instance of \( \text{R} \) is denoted by \( r := (t, c, e) \), where \( t \in T \) is a target of rule, \( c \in C \), where \( C = \{\text{Att}, \text{Op}, \text{V}\} \) denote the attribute of condition, operation and value, \( e \in E \), where \( E = \{D, P\} \) denote Permit and Deny wrt an effect. \( T = \{\text{subj}, \text{res}, \text{act}, \text{env}\} \) denote Subject, Resource, Action, and Environment attributes, where \( \text{rd} \) denotes “read”, \( \text{w} \) “write”, \( \text{del} \) “delete”, \( \text{app} \) “append” and \( \text{exe} \) “execute”.

The policy-set specifies the target that policy is applicable to and a set of policies, e.g. policy of hospital A. Each policy-set contains a policy combining algorithm that prioritises the policies’ combination function. Consequently, each policy in a policyset contains the combining algorithms for combining rules and a set of rules that define access rights. Users in the system can define their access privileges through a set of rules comprising the attributes of targets, the conditions of access to data and permit or deny access. For example, Rule \( R = \{(\text{FinanceStaff}, \text{Billinginfo}, \text{Read}, \text{HospitalA}), 8 < \text{Time} < 17, \text{Permit}\} \) denotes that the finance staff in HospitalA have the permission to read patient’s billing information during business hours.
5.4.2 Policy similarity

In [BDK+15] and [LRF+13], the similarity wrt attributes is computed in a hierarchical structure. To better serve our goal, we employ a linear equation to compute the similarity value, resulting in lower computation complexity. A case study is illustrated in next section to clarify EHR policy combination.

Definition 5.4 (Equal Policies) The similarity score reflects the similarity of the policies wrt rules. Policies that comprise similar targets and conditions and can be applied to same requests are called Equal Policies. Equal policies consist of similar targets and rules.

We compute the similarity score among two policies \( \text{pol} \in \text{Pol} \) and \( \text{pol}' \in \text{Pol}' \) with the following formula in which the target of policies are declared by rules:

\[
S(\text{pol}, \text{pol}') = \omega_p S_{\text{rule-set}}^P + \omega_d S_{\text{rule-set}}^D,
\]

\( S_{\text{rule-set}}^P \) and \( S_{\text{rule-set}}^D \) are the similarity scores of policies based on same effects (Deny, Permit), respectively. \( \omega_p \) is the importance weight for permit rules, \( \omega_d \) is the importance weight for deny rules and \( \omega_t \) is the importance weight of policy targets similarity, where \( \omega_p + \omega_d = 1 \). However, if there are any separated policy targets, the similarity would be computed as

\[
S(\text{pol}, \text{pol}') = \omega_t S_{\text{rule-set}}^T + \omega_p S_{\text{rule-set}}^P + \omega_d S_{\text{rule-set}}^D.
\]

We illustrate the similarity computation in the following section.

Similarity measure

In order to compute rules similarity, we map and compare rule \( r_i \in R \) of policy \( \text{pol} \) to a set of rules \( r'_j \in R' \) belonging to \( \text{pol}' \) of the same effect \( E \). As Permit and Deny rule sets are defined similarly, in the following, we only present a general case of computation, for either Permit or Deny rule set. This process is demonstrated as mapping one-to-many over the sum of all the mappings in the rule as given in Eq. (5.3) (illustrated in Algorithm 3 later):

\[
S_{\text{rule-set}} = \frac{\sum_{r_i \in R, r'_j \in R'} [S_{\text{rule}}(r_i, r'_j)]}{\max(|R|, |R'|)}
\]
5.4. Proposed policy combination system

$S_{\text{rule}}(r_i, r'_j)$ computes the similarity of rules $r_i$ and $r'_j$ based on rule targets and rule conditions in Eq. (5.4):

$$S_{\text{rule}}(r_i, r'_j) = \omega_t S_t(r_i, r'_j) + \omega_c S_c(r_i, r'_j), \quad (5.4)$$

where

$$S_t(r_i, r'_j) = a_1 s_{\text{subj}}(r_i, r'_j) + a_2 s_{\text{res}}(r_i, r'_j) + a_3 s_{\text{act}}(r_i, r'_j) + a_4 s_{\text{env}}(r_i, r'_j), \quad (5.5)$$

$$S_c(r_i, r'_j) = b_1 s_{\text{Att}}(r_i, r'_j) + b_2 s_{\text{Op}}(r_i, r'_j) + b_3 s_{\text{V}}(r_i, r'_j). \quad (5.6)$$

Here, $S_t(r_i, r'_j)$ denotes the similarity score of rule targets and $S_c(r_i, r'_j)$ denotes the similarity score of rules’ conditions, where $s_{\text{subj}}(r_i, r'_j)$, $s_{\text{res}}(r_i, r'_j)$, $s_{\text{act}}(r_i, r'_j)$ and $s_{\text{env}}(r_i, r'_j)$ are similarity scores of subject, resource, action and environment attributes, respectively. We require $a_1 + a_2 + a_3 + a_4 = 1$. $s_{\text{Att}}(r_i, r'_j)$, $s_{\text{Op}}(r_i, r'_j)$, $s_{\text{V}}(r_i, r'_j)$ are the similarity scores of Attr, Op and V, respectively. We require $b_1 + b_2 + b_3 = 1$. In addition, the importance weights must satisfy $\omega_t + \omega_c = 1$. The value of base similarity is ranged in $[0,1]$. In the following sections, we present the similarity computation of attributes and conditions.

We evaluate the attribute similarity among rules of different policies $\text{pol}$ and $\text{pol}'$ that results in the similarity score of the attributes $S_t(R)$ in Eq. (5.3) for computing rules similarity with the same effect, to clarify, $S_t(r_i, r'_j)$ results the mapping $r_i$ with $r'_j$.

Take an example. Assuming $a_1 = a_2 = a_3 = a_4 = 1/4$ and $s_{\text{subj}}(r_i, r'_j) = 2/5$, $s_{\text{res}}(r_i, r'_j) = 0$, $s_{\text{act}}(r_i, r'_j) = 2/3$, and $s_{\text{env}}(r_i, r'_j) = 0$, the similarity score of rule $r_1 \in \text{pol}_1$ and $r'_2 \in \text{pol}_2'$ is computed with Eq. (5.5) as following:

$$S_t(r_1, r'_2) = \frac{1}{4} \cdot \frac{2}{5} + \frac{1}{4} \cdot 0 + \frac{1}{4} \cdot \frac{2}{3} + \frac{1}{4} \cdot 0 = 0.27$$

**Definition 5.5** (Named Attribute) Let $\mathcal{Att} = \{a_1 \otimes v_1, \ldots, a_i \otimes v_i\}$ be a set of named attributes where $a_i$ is a binary attribute, $v_i$ is a value, and $\otimes : \{\leq, \geq, <, >, =, \neq\}$ is an operation over the set of attributes.

In XACML, named attribute $\mathcal{Att}$ is a specific instance of an attribute, determined by the attribute name and type, the identity of the attribute holder (which may be of type: subject, resource, action or environment) and (optionally) the identity of the issuing authority. In order to compute the similarity score of the rules’ conditions, we use three types of attributes as $\text{ConditionAtt}$, $\text{Operation}$ and $\text{Value}$ (Eq. (5.6)).
Provided that the conditions of rules \( r_i \) and \( r'_j \) are similar, the similarity score will be \( S_c = 1 \), otherwise \( 0 \leq S_c \leq 1 \). In particular, where either of rules involves a condition, restricting another rule, the \( S_c = 1/2 \).

For instance, we take two conditions for granted as \( r_1 : T \leq 9.00PM \) and \( r'_2 : 6.00PM \leq T \leq 9.00PM \) where the similarity can be computed with the condition similarity in Eq. (5.6).

\[
S_c(r_1, r'_2) = \frac{1}{3} \cdot 1 + \frac{1}{3} \cdot \frac{1}{2} + \frac{1}{3} \cdot \frac{1}{2} = \frac{2}{3}
\]

where we assume equal weights of the elements and \( s_{Alt}(r_i, r'_j) = 1 \), \( s_{Op}(r_i, r'_j) = 1/2 \), and \( s_V(r_i, r'_j) = 1/2 \).

In order to compute the similarity of rules, a comparison of rules’ components belonging to the same effects is essential. Given two policies \( \text{pol} \) and \( \text{pol}' \), our algorithm groups policy’ rules based on two different effects permit and deny shown as \( E : \{P, D\} \). We use numerical predicates to evaluate the similarity of rules based on same values. Each rule in \( \text{pol} \) is compared to a single rule (one-one mapping) in \( \text{pol}' \) belonging to the same effect in Eq. (5.3).

\[
S_{\text{rule-set}} = \frac{\sum_{r_i \in R, r'_j \in R'} [S_{\text{rule}}(r_i, r'_j)]}{\max(|R|, |R'|)}
\]

\[
= \frac{\frac{1}{2} \times 0.27 + \frac{1}{2} \times \frac{2}{3}}{1}
\]

\[
= 0.47
\]

which is assumed to be \( S^P_{\text{rule-set}} \), and \( S^D_{\text{rule-set}} \) can be computed in the same way. Say its value is the same. Targets of policies are the same as well. The final similarity result can then be computed with Eq. (5.2), where \( S^T_{\text{pol,pol}'} \) is the similarity score of the targets of policies (algorithm 3):

\[
S(\text{pol}, \text{pol}') = \omega_t S^T_{\text{pol,pol}'} + \omega_p S^P_{\text{rule-set}} + \omega_d S^D_{\text{rule-set}}
\]

\[
= \frac{1}{3} \times 1 + \frac{1}{3} \times 0.47 + \frac{1}{3} \times 0.47
\]

\[
= 0.64
\]

Algorithm 3 shows the process of rule similarity computation according to Eq. (5.3). Rules in policies \( P_1 \) and \( P_2 \) are compared according to same effects, deny or permit. Then, the total similarity is the addition of similarity of both policies according to policy targets and effects.
5.4. Proposed policy combination system

Let $r_D^i$ and $r_P^i$ denote the set of deny and permit rules in $P_1$.

Let $r_D^j$ and $r_P^j$ denote the set of deny and permit rules in $P_2$.

foreach $r_1 \in P_1$ and $r_2 \in P_2$ do $S_{rule}(r_1, r_2)$;

foreach $r_1 \in P_1^D$ and $r_2 \in P_2^D$ do $S_D^r(r_1, r_2)$;

foreach $r_1 \in P_1^P$ do $S^r(r_1, r_2)$;

foreach $r_2 \in P_2^P$ do $S^D(r_1, r_2)$;

if $r_1 \in P_1^P$, $r_j \in P_2^D$ then

foreach $r_1 \in P_1$, $r_j \in P_2$ do $S^D(r_1, r_j)$;

return $S^D(r_1, r_j)$

else if $r_1 \in P_1^D$, $r_j \in P_2^D$ then

foreach $r_1 \in P_1$, $r_j \in P_2$ do $S^D(r_1, r_j)$;

return $S^D(r_1, r_j)$

else return null;

for $S^r$ and $S^D$ do

$S_c(r_1, r_2)$;

end

return $S_{P_1, P_2} = \omega_r S^r(P_1, P_2) + \omega_p S^p(P_1, P_2) + \omega_d S^D(P_1, P_2)$;

Algorithm 3: RuleSimilarityMeasure Algorithm $(P_1, P_2)$

5.4.3 Policy combination

The similarity evaluation approach presented in the previous section allows the decision maker to decide if two or more policies should be combined. In the case of yes, these target policies should be combined based on the rule-combining algorithm and policy-combining algorithm provided in the XACML specification.

\[
\mathcal{CAP}(\text{pol}, \text{pol}^\prime) = S_{\text{rca}}(\text{pol}, \text{pol}^\prime) \times S(\text{pol}, \text{pol}^\prime)
\]  

(5.7)

$S_{\text{rca}}(\text{pol}, \text{pol}^\prime) \in \{0, \frac{1}{2}, 1\}$

Finally, the similarity value can be used in Eq. (5.7) to output the combination applicability value $\mathcal{CAP}(\text{pol}, \text{pol}^\prime)$:

\[
\mathcal{CAP}(\text{pol}, \text{pol}^\prime) = S_{\text{rca}}(\text{pol}, \text{pol}^\prime) \times S(\text{pol}, \text{pol}^\prime)
\]

\[
= 1 \times 0.64
\]

\[
= 0.64
\]

As it is shown in algorithm 4, the multiplication of $S_{\text{rca}}$ and similarity of both policies outputs $\mathcal{CAP}$:

Here, we adopt the similarity $S_{\text{rca}}(\text{pol}, \text{pol}^\prime)$ wrt the rule combination algorithm defined in XACML. We assume that pol and pol’ have adopted the XACML rule
5.4. Proposed policy combination system

Algorithm 4: Combination Applicability Value

combination algorithm, respectively. Our similarity evaluation takes them into account. In other words, we assume that the similarity evaluation also considers how the XACML combination algorithms are used in target policies. We will describe how to evaluate $S_{rca}$ in the next section.

Combining Algorithms

Recall that XACML contains a set of Policy, PolicySet and Rule elements with the rule-combining algorithm and policy-combining algorithm for combining the results of their evaluation process used by PDP unit. Combining algorithms can be declared by a RuleCombiningAlgId or PolicyCombiningAlgId attribute of Policy or PolicySet elements. To be precise, the PolicySet element contains a set of Policy elements and a policy-combining algorithm for combining a set of policies. Likewise, Policy element contains a set of Rule elements and a rule-combining algorithm for combining a set of rules. The XACML combining algorithm rules include, Deny-overrides (Ordered and Unordered), Permit-overrides (Ordered and Unordered), First-applicable and Only-one-applicable. The $S_{rca}$ can be specified in Tables 5.1 to 5.4 according to $rca$ (Algorithm 5).

Deny-overrides: Let $P, D, NA,$ and $Ud$ denote Permit, Deny, Non-applicable, and Undecidable, respectively. Deny-overrides (Table 5.1) says $D > P > NA > Ud$, which means that the evaluation result of a single Rule or Policy is Deny, regardless of the other Rule or Policy. Then, the $S_{rca}$ is 1 when there is one $D$, overriding the other rule’s effect. Similarly, Permit-overrides (Table 5.2)($P > D > NA > Ud$) indicates that the evaluation result of a single Rule or Policy will be Permit if a single Permit is appeared. Then $S_{rca}$ is 1 when there is one $P$, overriding the other rule’s effect.

First-applicable (Table 5.3) combining algorithm outputs the result of evaluating the first Rule or Policy element, which has applicable target and condition to the decision request. We consider $S_{rca} = 1$ when $pol_{rca} = pol'_{rca}$, and $S_{rca} = 1/2$ when
5.4. Proposed policy combination system

Algorithm 5: Combination Similarity

Table 5.1: Deny Override (ordered and unordered)

<table>
<thead>
<tr>
<th>pol\pol</th>
<th>P</th>
<th>D</th>
<th>NA</th>
<th>Ud</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>P (1)</td>
<td>D (1)</td>
<td>P (1/2)</td>
<td>Un (0)</td>
</tr>
<tr>
<td>D</td>
<td>D (1)</td>
<td>D (1)</td>
<td>D (1)</td>
<td>D (1)</td>
</tr>
<tr>
<td>NA</td>
<td>P (1/2)</td>
<td>D (1)</td>
<td>NA (0)</td>
<td>Ud (0)</td>
</tr>
<tr>
<td>Ud</td>
<td>Ud (0)</td>
<td>D (1)</td>
<td>Ud (0)</td>
<td>Ud (0)</td>
</tr>
</tbody>
</table>

both effects are applicable. Only-one-applicable (Table 5.4) only applies to policies, which means that only one policy or policy-set is applicable. If none of the policies is applicable, the result will be NotAplicable, \( S_{pca} = 0 \). On the other hand, the result will be Undecidable if more than one policy is applicable. When one policy applies, \( S_{pca} = 1/2 \), otherwise, \( S_{pca} = 0 \) (Tables 5.1-5.4).

To clarify, “permit-overrides rule-combining algorithm” means that if any rule evaluates to “True”, the access is permitted, otherwise, it returns “NotApplicable”. In the case of an error, the rule evaluation will return “Undecidable”. The rule-combining algorithm therefore specifies the combination of rules values into a single policy value.
5.5 A Case study

In this section, we provide a case study of policy combination in an EHR system. Take as an example where in hospital A, which offers a variety of healthcare services, both patient and policy owner (or system administrator) defined two types of privacy policies. In this case, they defined their own preferred privacy policies. As a consequence, in a case of conflicts of two established access control policies, patient data will not be accessible to caregivers. The patient denies any “write” and “delete” access permissions to its patient history by staff and doctors (GPs), whereas policy custodian permits only “read” and “write” access rights. The patient’s billing information cannot be deleted or changed by either patient or staff, then any changes are denied. Under no circumstances can billing information be accessible out of business hours. In addition, a patient permits “read”, “write” and “delete” access rights to laboratory, GP and Pharmacist.

In the best case scenario, policy combination process starts with assessing rule-combining algorithms, which is deny-overrides here. Policies can then go through the rule-similarity analysis phase to compute the similarity score among rules, show the probability of combining two policies. In Table 5.5 and 5.6, we provide the example of \( \text{pol} \) and \( \text{pol}' \) written in XACML. Based on them, we calculate the similarity scores. The similarity of both policies is analysed and then combined according to

<table>
<thead>
<tr>
<th>( \text{pol} ) ( \text{\textbackslash pol}' )</th>
<th>P</th>
<th>D</th>
<th>NA</th>
<th>Ud</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>P (1)</td>
<td>P (1)</td>
<td>P (1)</td>
<td>P (1)</td>
</tr>
<tr>
<td>D</td>
<td>P (1)</td>
<td>D (1)</td>
<td>D (1/2)</td>
<td>D (1/2)</td>
</tr>
<tr>
<td>NA</td>
<td>P (1)</td>
<td>D (1/2)</td>
<td>NA (0)</td>
<td>Ud (0)</td>
</tr>
<tr>
<td>Ud</td>
<td>P (1)</td>
<td>D (1)</td>
<td>Ud (0)</td>
<td>Ud (0)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( \text{pol} ) ( \text{\textbackslash pol}' )</th>
<th>P</th>
<th>D</th>
<th>NA</th>
<th>Ud</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>P (1)</td>
<td>P (1/2)</td>
<td>P (1/2)</td>
<td>P (1/2)</td>
</tr>
<tr>
<td>D</td>
<td>D (1/2)</td>
<td>D (1)</td>
<td>D (1/2)</td>
<td>D (1/2)</td>
</tr>
<tr>
<td>NA</td>
<td>P (1/2)</td>
<td>D (1/2)</td>
<td>NA (0)</td>
<td>Ud (0)</td>
</tr>
<tr>
<td>Ud</td>
<td>Ud (0)</td>
<td>Ud (0)</td>
<td>Ud (0)</td>
<td>Ud (0)</td>
</tr>
</tbody>
</table>
the proposed combination framework. As the result of CAP shows, policies \( \text{pol}_1 \) and \( \text{pol}_2 \) have the similarity score over 0.5 and could be combined. Note that the score of similarity is only a reference for the decision maker to decide whether they should be combined or not.

1. We categorise rules in policies \( \text{pol}_1 \), \( \text{pol}_2 \) based on their effects and find rules with the same effects. Here, superscribes \( P \) and \( D \) denote Permit and Deny, respectively.

\[
\text{pol}_1^P : \{r_{11}\}, \text{pol}_1^D : \{r_{12}\} \\
\text{pol}_2^P : \{r_{21}\}, \text{pol}_2^D : \{r_{22}\}, \text{pol}_2^D : \{r_{23}\}
\]

2. We compute the rule similarity scores between pairs of rules belong to the same effects based on a set of attributes. Then, we compute the similarity of conditions and targets between rules with same effects.

\[
\begin{align*}
S_c(r_{11}, r_{21}) &= 0 \\
S_c(r_{12}, r_{22}) &= [(\frac{1}{4} \times \frac{1}{2}) + (\frac{1}{4} \times \frac{1}{2}) + (\frac{1}{4} \times \frac{1}{2})] = 0.5 \\
S_c(r_{11}, r_{21}) &= 0 \\
S_c(r_{12}, r_{22}) &= [(\frac{1}{4} \times \frac{1}{2}) + (\frac{1}{4} \times \frac{1}{2}) + (\frac{1}{4} \times 1)] = 0.52 \\
S_c(r_{12}, r_{22}) &= [(\frac{1}{4} \times \frac{1}{2}) + (\frac{1}{4} \times \frac{1}{2}) + (\frac{1}{4} \times 1)] = 0.62 \\
S_c(r_{12}, r_{22}) &= [(\frac{1}{4} \times \frac{1}{2}) + (\frac{1}{4} \times \frac{1}{2}) + (\frac{1}{4} \times 1)] = 0.56
\end{align*}
\]

3. We compute rule similarity score based on similar targets and conditions.

\[
\begin{align*}
S(r_{11}, r_{21}) &= \frac{1}{2}[S_c(r_{11}, r_{21}) + S_c(r_{11}, r_{21})] = [\frac{1}{2} \times 0.52 + \frac{1}{2} \times 0] = 0.26 \\
S(r_{12}, r_{22}) &= \frac{1}{2}[S_c(r_{12}, r_{22}) + S_c(r_{12}, r_{22})] = [\frac{1}{2} \times 0.87 + \frac{1}{2} \times 0.5] = 0.68 \\
S(r_{12}, r_{22}) &= \frac{1}{2}[S_c(r_{12}, r_{22}) + S_c(r_{12}, r_{22})] = [\frac{1}{2} \times 0.62 + \frac{1}{2} \times 0.5] = 0.56
\end{align*}
\]

4. For each rule \( r_i \) in policy \( \text{pol}_1 \), we compute the similarity to a set of rules in \( \text{pol}_2 \).
5.5. A Case study

Table 5.5: The instances of \(pol_1\) and \(pol_2\) written with XACML.

<table>
<thead>
<tr>
<th>Policy 1: System Administrator</th>
<th>Policy 2: Patient</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Target&gt;</td>
<td>&lt;Target&gt;</td>
</tr>
<tr>
<td>&lt;Rule RuleId=“R11” Effect=“Permit” &gt;</td>
<td>&lt;Rule RuleId=“R21” Effect=“Permit” &gt;</td>
</tr>
<tr>
<td>&lt;Target&gt;</td>
<td>&lt;Target&gt;</td>
</tr>
<tr>
<td>&lt;Subject Designation belong to {GP, Nurse, Pharmacist, Staff} &gt;</td>
<td>&lt;Subject Designation belong to {Laboratory, GP, Pharmacist} &gt;</td>
</tr>
<tr>
<td>&lt;Action AccessType belong to {Read, Write} &gt;</td>
<td>&lt;Action AccessType belong to {Read, Write, Delete} &gt;</td>
</tr>
<tr>
<td>&lt;Resource FileType belong to {PatientHistory} &gt;</td>
<td>&lt;Resource FileType belong to {Prescription} &gt;</td>
</tr>
<tr>
<td>&lt;Condition 8 &lt;= Time &lt;= 17 &gt;</td>
<td>&lt;Condition FileSize &gt; 300MB &gt;</td>
</tr>
<tr>
<td>&lt;/Target&gt;</td>
<td>&lt;/Target&gt;</td>
</tr>
<tr>
<td>&lt;Rule RuleId=“R12” Effect=“Deny” &gt;</td>
<td>&lt;Rule RuleId=“R22” Effect=“Deny” &gt;</td>
</tr>
<tr>
<td>&lt;Target&gt;</td>
<td>&lt;Target&gt;</td>
</tr>
<tr>
<td>&lt;Subject Designation belong to {Staff} &gt;</td>
<td>&lt;Subject Designation belong to {Staff, Patient} &gt;</td>
</tr>
<tr>
<td>&lt;Resource FileType belong to {BillingInfo} &gt;</td>
<td>&lt;Resource FileType belong to {BillingInfo} &gt;</td>
</tr>
<tr>
<td>&lt;Action AccessType belong to {Write, Delete} &gt;</td>
<td>&lt;Action AccessType belong to {Write, Delete} &gt;</td>
</tr>
<tr>
<td>&lt;/Target&gt;</td>
<td>&lt;/Target&gt;</td>
</tr>
<tr>
<td>&lt;Rule RuleId=“R23” Effect=“Deny” &gt;</td>
<td>&lt;Rule RuleId=“R23” Effect=“Deny” &gt;</td>
</tr>
<tr>
<td>&lt;Target&gt;</td>
<td>&lt;Target&gt;</td>
</tr>
<tr>
<td>&lt;Subject Designation belong to {Staff, GP} &gt;</td>
<td>&lt;Subject Designation belong to {Staff, GP} &gt;</td>
</tr>
<tr>
<td>&lt;Resource FileType belong to {PatientHistory} &gt;</td>
<td>&lt;Resource FileType belong to {PatientHistory} &gt;</td>
</tr>
<tr>
<td>&lt;Action AccessType belong to {Write, Delete} &gt;</td>
<td>&lt;Action AccessType belong to {Write, Delete} &gt;</td>
</tr>
<tr>
<td>&lt;Condition FileSize &gt; 300MB &gt;</td>
<td>&lt;Condition FileSize &gt; 300MB &gt;</td>
</tr>
<tr>
<td>&lt;/Target&gt;</td>
<td>&lt;/Target&gt;</td>
</tr>
<tr>
<td>&lt;/Policy&gt;</td>
<td>&lt;/Policy&gt;</td>
</tr>
</tbody>
</table>

\[ S_{pol_1, pol_2}^{pol_1, pol_2} = S_{pol_1, pol_2} : S(r_{11}, r_{21}) = 0.26 \]
\[ S_{pol_1, pol_2}^{pol_1, pol_2} : S(r_{22}, r_{12}) = 0.68 \]
\[ S_{pol_1, pol_2}^{pol_1, pol_2} : S(r_{23}, r_{12}) = 0.56 \]
\[ S_{pol_1, pol_2}^{pol_1, pol_2} : \frac{1}{2}[S(r_{12}, r_{22}) + S(r_{12}, r_{23})] = \left[ \frac{1}{2} \times 0.68 + \frac{1}{2} \times 0.56 \right] = 0.62 \]

5. The average of permit rules similarity and deny rule similarities with similarity of conditions are computed as following:
\[ S^P(pol_1, pol_2) = \frac{1}{2}[S_{pol_1, pol_2}^{pol_1, pol_2} + S_{pol_1, pol_2}^{pol_1, pol_2}] = \left[ \frac{1}{2} \times 0.26 + \frac{1}{2} \times 0.26 \right] = 0.26 \]
\[ S^D(pol_1, pol_2) = \frac{1}{3}[S_{pol_1, pol_2}^{pol_1, pol_2} + S_{pol_1, pol_2}^{pol_1, pol_2} + S_{pol_1, pol_2}^{pol_1, pol_2}] = \left[ \frac{1}{3} \times 0.62 + \frac{1}{3} \times 0.68 + \frac{1}{3} \times 0.56 \right] = 0.62 \]

6. The similarity among each pair of policies is computed as following:
\[ S(pol_1, pol_2) = \frac{1}{3}[S^P(pol_1, pol_2) + S^D(pol_1, pol_2) + S^P(pol_1, pol_2)] = \left[ \frac{1}{3} \times 1 + \frac{1}{3} \times 0.62 + \frac{1}{3} \times 0.26 \right] = 0.63 \]

7. We compute the combination value to find the applicability of policies combination.
Table 5.6: Combined policies

```
<PolicySet>
  <PolicySetId="PolicySet1"/>
  <PolicyCombiningAlgId="identifier:policy-combining-algorithm:deny-overrides"/>
  <Policy>
    <PolicyId="Policy1"/>
    <RuleCombiningAlgId="identifier:rule-combining-algorithm:deny-overrides"/>
    <PolicyTarget> Hospital A </PolicyTarget>
    <Rule>
      <RuleId="RC1"/>
      <Effect>Permit</Effect>
      <Target>
        <Subject Designation belong to {Pharmacist, Nurse, Staff}>
        <Resource FileType belong to {PatientHistory}>
        <Action AccessType belong to {Read, Write}>
      </Target>
    </Rule>
    <Rule>
      <RuleId="RC2"/>
      <Effect>Permit</Effect>
      <Target>
        <Subject Designation belong to {Laboratory, GP, Pharmacist}>
        <Resource FileType belong to {Prescription}>
        <Action AccessType belong to {Read, Write, Delete}>
      </Target>
    </Rule>
    <Rule>
      <RuleId="RC3"/>
      <Effect>Deny</Effect>
      <Target>
        <Subject Designation belong to {Staff, patient}>
        <Resource FileType belong to {BillingInfo}>
        <Action AccessType belong to {Write, Delete}>
        <Condition 8 <= Time <= 17>
      </Target>
    </Rule>
    <Rule>
      <RuleId="RC4"/>
      <Effect>Deny</Effect>
      <Target>
        <Subject Designation belong to {Staff, GP}>
        <Resource FileType belong to {PatientHistory}>
        <Action AccessType belong to {Delete, Write}>
        <Condition FileSize > 300MB>
      </Target>
    </Rule>
  </Policy>
</PolicySet>
```

\[ CAP(\text{pol}_1, \text{pol}_2) = S_{src}(\text{pol}_1, \text{pol}_2) \times S(\text{pol}_1, \text{pol}_2) = 1 \times 0.63 = 0.63 \]

5.6 Conclusion

In this chapter, we introduced the policy similarity to deal with multiple access control policies, which adds a layer of functionality to implementing the XACML system. A similarity value specifies the applicability of policies combination. Our approach provides a solution for the decision maker to decide whether two or more XACML policies should be combined, based on the value of policy similarity. Our proposed tool can be placed on the top of these algorithms. Our access control policy combination solution has been shown useful in our case study in order to preserve the privacy of users and ensure data security.
Chapter 6

Practical and Secure Telemedicine Systems for User Mobility

The application of wireless devices has led to a significant improvement in the quality delivery of care in telemedicine systems. Patients who live in a remote area are able to communicate with the healthcare provider and benefit from the doctor consultations. However, it has been a challenge to provide a secure telemedicine system, which captures users (patients and doctors) mobility and patient privacy. Our work addresses these issues. Owing to the importance of patient’s information and communication channels prone to different attacks, protecting patient records becomes an inevitable necessity. In this chapter, we present secure protocols for telemedicine systems, which ensure the secure communication between patients and doctors who are located in different geographical locations. Our protocols are the first of this kind featured with confidentiality of patient information, mutual authentication, patient anonymity, data integrity, freshness of communication and mobility. Our security protocols are symmetric-key based in order to better serve our objectives of research for secure telemedicine services. We present four different scenarios which cover almost all common situations for practical applications.

6.1 Introduction

In traditional medical systems, patients have to visit a clinic or a hospital to have a doctor consultation and treatment; therefore, it is inconvenient for elderly patients and patients residing in rural and distant areas, especially for those with chronic diseases. Thanks to Telecare Medical Information Systems (TMIS) which have made outstanding advances and provided efficient communications among patients and doctors. Although many countries are demanding home-based long-term care in order to provide efficient treatment for patients and increase their quality of lives, it
also arises some security and privacy concerns among healthcare providers and users in terms of disclosing patient information in insecure communication channels.

The growth of wireless network devices in medical services has been in an inevitable pace to offer various applications in healthcare. There are many studies addressing the security pitfalls of TMIS through various methods; however, there are still many security issues and implementation obstacles. Medical information should be protected through authenticated channels and cryptographic solutions. There are many works in the literature, which have proposed various solutions. However, it is still a challenge to provide an efficient and secure system which captures all necessary needs for remote medical services, including mobility of patients and doctors as well as the anonymity of patients against disclosure of patient identities.

In this work, for the first time, we capture all the aforementioned features for practical and ideal telemedicine service. As the main contribution, we systematically studied how to secure the wireless medical systems while patients and doctors can be mobile and in the meanwhile, patients can still remain anonymous. We overcome the security hurdle due to the mobility of patients and doctors. We present our approaches with four application scenarios in terms of the mobility and show that in those scenarios, patients and doctors can establish a secure communication channel which meets all our security and privacy requirements, i.e., patient’s privacy (anonymity), communication freshness, data confidentiality and data integrity.

Before presenting our protocols, we review related work and show the research gap that our work will fill.

In the literature, many security schemes have been proposed to ensure secure communication. These studies employed user authentication and session key agreement protocols \[\text{AIB}^+15, \text{KKLW}16\]. However, the proposed solutions suffer from different types of attacks, insider attacks, anonymity problems, replay attacks, etc \[\text{WLL}^+12b, \text{HCZ}12, \text{WHL}12, \text{JMML}13, \text{JMLT}14, \text{MSM}14\].

We noticed that there are some works which have considered confidentiality and authentication in telemedicine (or can be potentially applied to telemedicine). For instance, Yang et al. \[\text{YKM}15\] proposed a privacy-preserving authentication scheme with adaptive key evolution. A similar work proposed by Chen et al. \[\text{CLC}16\] discussed multi-channel safety authentication protocols in wireless networks. Jiang et al. \[\text{JLY}^+16\] also proposed an authentication scheme for wireless body area networks in mhealth. However, they do not address the mobility of patients and
There are further proposed solutions for telemedicine systems offering authentication schemes [HCZ12, WHL12, JMML13, JMLT14], which provide authentications services for patients, healthcare server, doctors, along with the security analysis on impersonation attack, replay attack, message authenticity, backward/forward secrecy and confidentiality. In addition, some other studies provide biometric-based authentication schemes [MSM14]. In the literature, gateways are designated to register patients and doctors, which can threat the security if an attacker eavesdrops the intermediate point. Rahman et al. [RMH+16] also considered patient privacy using an attribute-based setting.

We compare the security properties of our protocols with other proposed solutions in Table 6.1. As it is clearly stated, aforementioned studies in the literature offer some security properties which have been mentioned; however, they have only captured some basic security properties without considering mobility. Our work is the only work which captures this important feature for telemedicine.

<table>
<thead>
<tr>
<th>Security properties</th>
<th>[AIB+15]</th>
<th>[WLL+12b]</th>
<th>[HCZ12]</th>
<th>[WHL12]</th>
<th>[JMML13]</th>
<th>[YKM15]</th>
<th>[JLY+16]</th>
<th>Our protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confidentiality</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Authentication</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Anonymity</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Mobility</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 6.1: Comparison of the proposed scheme with related works.

In the following sections, we first present our system model and our security assumptions (Section 6.2). Based on them we demonstrate four practical telemedicine scenarios and their secure communication protocols and routing tables in Section 6.3 and 6.4, along with two clinical examples in Section 6.5. We also provide the security analysis in Section 6.6. We conclude the chapter in Section 6.7.

6.2 System Model and Security Requirements

6.2.1 System Model

We consider a general case in a medical system, which consists of a healthcare centre, patients and doctors. Consider a scenario of telemedicine, it consists of a healthcare provider, authentication server, patients and doctors. For simplicity and clearness of presentation, we refer to “authentication server” as an entity which
represents healthcare provider, system administration and security service. Patients and doctors are registered in the authentication server.

![System Model](image)

Figure 6.1: System model.

We provide a solution to telemedicine which can be wireless and wired. Our solution captures security and privacy requirements mentioned earlier along with mobility of users. In Fig. 6.1, we illustrate our system model, where the healthcare centre provides the entire service, users (doctors and patients) are located at different regions or domains, which are serviced by authentication servers (local or remote), depending on the user location. In our model, we allow users to move to a different domain dynamically while maintaining their ability to communicate securely with each other. Our system captures all possible situations and provides the same level of security services to users, regardless of the type of location.

### 6.2.2 Security Requirements and Notations

We presume that users (patients and doctors) belong to their home-area network, in which they are registered. Users share a long-term secret key with their home server. The setup of the key is done at the registration. Our system only requires symmetric keys, which offer much better computational efficiency. We consider the following security assumptions for our protocols:

- Confidentiality. We assume that all users share a long-term key with their
home server (or authentication server). With this as the basis, we can construct secure channels against eavesdropping from outsiders. Here, by outsiders, we mean that anyone who is not registered with the system and those who have registered with the system but are not involved in the protocol execution.

- **Anonymity.** We require all patients’ identities to be protected against outsiders.

- **Authentication.** We require mutual authentication between patient and servers as well as patient and doctor. We achieve mutual authentication by allowing trust between users and their home server as well as trust amongst all servers.

- **Freshness.** Freshness against replay attacks from outsiders. We achieve this feature by using nonces. Notice that we can add timestamps to our protocols to secure them against suppress attacks; however, for simplicity we omit it. We only use the timestamp for service tickets.

- **Integrity.** Integrity against all outsiders is achieved with message authentication code (MAC).

In Table 6.2, we provide the notations used in the description of the protocols.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_i$</td>
<td>the $i$-th user’s ID (a user could be a patient $P_i$ or a doctor $D_i$)</td>
</tr>
<tr>
<td>$U_{ij}$</td>
<td>the $i$-th user’s $j$-th subliminal ID</td>
</tr>
<tr>
<td>HS</td>
<td>Home-server</td>
</tr>
<tr>
<td>RS</td>
<td>Remote-server</td>
</tr>
<tr>
<td>$K_{u,v}$</td>
<td>Shared secret-symmetric key between party $u$ and $v$</td>
</tr>
<tr>
<td>$k_s$</td>
<td>Secret session key</td>
</tr>
<tr>
<td>$T$</td>
<td>Time stamp</td>
</tr>
<tr>
<td>$[\text{Data}]_k$</td>
<td>Data encrypted with a symmetric key $k$</td>
</tr>
<tr>
<td>$h(\cdot)$</td>
<td>A secure cryptographic hash function</td>
</tr>
<tr>
<td>$n_u$</td>
<td>Nonce generated by user $U$</td>
</tr>
<tr>
<td>$U_i \rightarrow U_j$</td>
<td>message: $U_i$ sends message to $U_j$</td>
</tr>
</tbody>
</table>

Table 6.2: Notations

### 6.2.3 System Setup

There are two types of servers in our system model, including home server and remote server.
6.2. System Model and Security Requirements

- **Home Server** (HS): HS is situated in the home-area network that stores the real identity of user U, who can be a doctor (D) or a patient (P), and a long-term secret-symmetric key $K_{u,hs}$, which is shared by a user and HS. HS also maintains a routing table ($\text{Tab}_{hs}$) which stores the real identity of the user and the corresponding subliminal identity and provides the mapping of them. It also stores a long-term secret key $K_{hs,rs}$ that is used to communicate between the home server HS and the remote server RS.

- **Remote Server** (RS): RS is situated in the remote-area network and stores a long-term secret key $K_{hs,rs}$ that is used to communicate between HS and RS and a long-term secret key $K_{ru,rs}$, which is shared by the remote user and RS. To remote users, RS is their HS, if they have registered at the RS as their home server. Similarly to the HS, the RS maintains a routing table ($\text{Tab}_{rs}$) to service its registered users.

All patients and doctor need to register with their home server in order to receive the service.

- **Patient Registration**: The registration phase is a one-time process between a patient and its home server HS. The purpose of patient registration is to setup the shared long-term secret key and register its identity. In turn, the patient will receive a subliminal identity and the secret key $K_{p,hs}$. As an option, $K_{p,hs}$ can be derived from the patient’s password. In this case, the patient should setup its password during the registration phase. The HS will update its table which stores the mappings of real identity and subliminal identity.

- **Doctor Registration**: Only difference between patient registration and doctor registration is that the doctor is not anonymous, i.e., there is no need to assign a subliminal identity to a doctor. Therefore, the doctor identity is registered with its HS and obtains a long-term secret key shared with the HS. As an option, $K_{d,hs}$ can be derived from the doctor’s password. In this case, the doctor should setup its password during the registration phase.
6.3 Protocols

We consider four practical scenarios, for a patient to establish a secure communication channel with its doctor who is located at the same or a different location. They communicate by an established wireless or wired communication system. The protocols for other scenarios can be easily obtained with the protocols for these four typical scenarios.

6.3.1 Scenario 1

We consider a scenario (Fig. 6.2) where a patient who is situated in its own local place needs to consult with a doctor due to its illness. In this scenario, both the patient and the doctor reside in the same communication domain, which is the home for both of them and is managed by HS.

![Image](image.png)

Figure 6.2: Scenario 1. The figure on the left-hand side is for the patient ticket granting. The figure on the right-hand side is for the consultation session.

1. Obtaining a ticket.

The objective of this phase is to establish a secure communication channel between a patient and a doctor. This process requires mutual authentication between the patient and the doctor. This is done with the help of the HS who is trusted by the patient and the doctor. The output from this phase is a session key shared by the patient and the doctor. The following protocol is conducted by patient $P$ and home-sever $HS$ in order to obtain a service from doctor $D$. The patient $P$ has a subliminal identity: $P_i$. For simplicity, we have omitted the subscript $i$. 
6.3. Protocols

(a) \( P \rightarrow HS : P_1^*, HS, D, n_p, MAC_{K_{p,hs}}(P_1^*, HS, D, n_p) \)

(b) \( HS \rightarrow P : HS, P_1^*, n_p, [k_s, P_2^*]_{K_{p,hs}}, Ticket, MAC_{K_{p,hs}}(HS, P_1^*, n_p, k_s, P_2^*, Ticket) \)

By Ticket_{p,d}, we denote the ticket for the patient to communicate with the doctor, where

\[ Ticket = [k_s, P, P_1^*, D, T]_{K_{d,hs}} \]

With the Ticket_{p,d}, the patient P can contact the doctor for consultation.

2. Remote consultation phase.

In this phase, the patient P communicates with the doctor D by sending Ticket_{p,d} to D. In turn, the doctor D replies to the patient P to confirm the establishment of the communication channel. The communication before P and D is conducted via the home server HS. Therefore, the subliminal identity of P can still be used to protect the patient. For simplicity, we omit the intermedia steps.

(a) \( P \rightarrow (HS) \rightarrow D : P_1^*, D, n_p, Ticket, MAC_{k_s}(P_1^*, D, n_p, Ticket) \)

(b) \( D \rightarrow (HS) \rightarrow P : D, P_1^*, n_p, ok, MAC_{k_s}(D, P_1^*, n_p, ok) \)

(c) \( P \rightarrow (HS) \rightarrow D : \cdots \) Consultation request

(d) \( D \rightarrow (HS) \rightarrow P : \cdots \) Reply

This communication channel can last while there is a necessity. All the communication messages are encrypted with session key \( k_s \), which provides the secure communication channel.

6.3.2 Scenario 2

We consider a scenario (Fig. 6.3) in which one of the patients (e.g. P) has travelled to another domain, rather than its own local area network, and requests a communication channel for the remote consultation session with the doctor who is still located at its home, which is managed by HS. Differing from the scenario 1, the patient P is unable to contact its home server HS directly. The communication with HS must be mediated by the visiting remote server RS, which does not share any secret key with P. Since HS and RS share a long-term secret key, the authentication communication flows are passed by RS to the home-server HS who then authenticates P in order to establish a consultation session with the doctor D.
6.3. Protocols

Figure 6.3: Scenario 2. The figure on the left-hand side shows the case when the patient has travelled to the remote domain, while the doctor is still in its home domain. The figure on the right-hand side shows the consultation phase, which is done with the assistance of HS and RS.

1. Obtaining a ticket.

In order to establish a secure communication channel between patient and doctor (similar to Scenario 1), the patient needs to contact their home server to obtain a service ticket. In this scenario, the patient is located at the remote domain (RS) and needs to contact the remote server RS that acts as a mediator. The mutual authentication between P and D is done with the help of RS and HS, where the service ticket plays an important role. The main hurdle for this protocol to work is how to let the RS check whether P is a legitimate user. Since the RS does not hold any information about P, then it is unable to verify the authenticity of P, with the identification information provided by P. We solve this problem by allowing the P to encrypt the data with a temporary session key. Later, the HS can help provide such a key to the RS, therefore, the RS can check the authenticity of the P.

(a) \( P \to RS : P^*_i, RS, D, HS, n_p, Token, MAC_{tk}(P^*_i, RS, D, HS, n_p, Token) \)

The patient P contacts the remote server RS as the first step to obtaining a service ticket. Here, \( Token = [P^*_i, HS, RS, n_p]_{K_{p,hs}} \) and \( tk \) is a temporary key for the RS to verify the MAC after it has obtained the key from the HS. \( tk \) is derived from \( tk = f(K_{p,hs}, n_p, P) \), where \( f(\cdot) \) is a cryptographic hash function.

(b) \( RS \to HS : RS, HS, P^*_i, D, n_p, n_{rs}, Token, MAC_{K_{rs,hs}}(RS, HS, P^*_i, D, n_p, n_{rs}, Token) \)
In this step, RS forwards the Token to HS in order to get the temporary key \( tk \). HS can decrypt Token and check the authenticity of P, and then compute the temporary key \( tk = f(K_{p,hs}, n_p, P) \).

(c) HS → RS : HS, RS, \( P^*_1 \), \( [P'_1, tk, n_p, n_{rs}]_{K_{rs,hs}}, MAC_{K_{rs,hs}}(HS, RS, P^*_1, tk; n_p, n_{rs}) \), Package

where

\[
\text{Package} = [P'_2, D, k_s, \text{Ticket}]_{K_{p,hs}}, \\
\text{Ticket} = [P, P^*_1, D, k_s, T]_{K_{d,hs}}
\]

Two main tasks in this step are: (1) HS needs to forward \( tk \) to RS. This is done by encrypting it along with the subliminal ID of P and nonces. \( n_{rs} \) is initialised by RS, hence it must be returned to RS to indicate the completion of the session between RS and HS. \( n_p \) is initialised by P, so it must be forwarded to RS so that it can be returned to P in the next step. (2) Package is sent to RS, so that it can forward it to P in the next step. The Package contains Ticket which is sent to P in the next step. In order for P and D to establish a secure channel for the consultation, a session key \( k_s \) is embedded in Package, to ensure both P and D will obtain it. The doctor D will obtain \( k_s \) from the service ticket Ticket when P requests a consultation session to D.

(d) RS → P : RS, \( P^*_1 \), \( n_p \), MAC\(_{tk}(RS, P^*_1; n_p) \), Package

In this step, Package is delivered to P by RS. P therefore obtains the service ticket Ticket and the secret session key \( k_s \); hence P is ready to contact his doctor D for consultation. As a necessary matter, P needs to check if \( n_p \) is the same as the one initialised by itself in the first step. If the checking step returns true, the entire protocol run is complete.

2. Remote consultation phase.

With the service ticket Ticket, P can then contact its doctor D. The protocol is the same as that in the first scenario. The only difference is that the communication between P and D is conducted with the aid of both RS and HS.
6.3.3 Scenario 3

In this case (Fig. 6.4), both patient P and doctor D have left their home domain and are situated in a distant location. The phase of ticket granting phase for the patient P is the same as that in the second scenario. However, since the doctor D has also left its home domain and is located at the remote domain managed by RS, the D needs to be authenticated by HS. Therefore, in this case, we only present this part of the protocol.

![Diagram of Scenario 3](image)

Figure 6.4: Scenario 3. The figure on the left-hand side shows that the patient and the doctor both have left HS and are now located at the remote domain managed by RS. The figure on the right-hand side shows the consultation communication flows.

In order to establish a secure communication channel between patient and doctor, the patient and the doctor need to contact their home server HS, since the remote server RS does not store any information about them. The mutual authentication among P and HS is done with the help of RS and HS, as the protocol given in Scenario 2. Accordingly, D needs to be authenticated by RS in order to establish a communication channel with its patients. Since RS does not hold any information about D, then it can verify the legitimacy of D only with the help of HS who registered D. The authentication protocol for D is similar to the patient authentication, while D does not require a subliminal ID and a service ticket.

1. Authentication phase.

   The doctor authentication protocol is given as follows:

   \[(a) \quad D \rightarrow RS : D, RS, HS, n_d, Token, MAC_{tk}(D, RS, HS, n_d, Token)\]

   The doctor D contacts the remote server RS as the first step in order to be authenticated. Here, Token = \[|D, HS, RS, n_d|_{K_{d,hs}}\] and \(tk\) is a temporary
key for the RS to verify the MAC after it has obtained the key from the HS. \(tk\) is derived from \(tk = f(K_{d,hs}, n_d, D)\), where \(f(\cdot)\) is a cryptographic hash function.

(b) RS \(\rightarrow\) HS : RS, HS, D, \(n_d, n_{rs}\), Token, MAC\(_{K_{rs,hs}}\)(RS, HS, D, \(n_d, n_{rs}\), Token)

In this step, RS forwards the Token to HS in order to get the temporary key \(tk\). The HS also checks the D’s information stored in HS in order to authenticate D.

(c) HS \(\rightarrow\) RS : HS, RS, [D, \(tk, n_d, n_{rs}\)]\(_{K_{rs,hs}}\), MAC\(_{K_{rs,hs}}\)(HS, RS, D, \(tk, n_d, n_{rs}\))

This step captures the following task. HS forwards \(tk\) to RS in order to verify the authenticity of D by RS. This is done by encrypting it along with the ID of D and nonces. RS can now verify the MAC provided by D in the first step. This step also confirms that D is a legitimate user of HS.

(d) RS \(\rightarrow\) D : D, RS, \(n_d\), accept/reject, MAC\(_{tk}\)(RS, D, \(n_d\), accept/reject)

The protocol ends after RS confirms “accept” or “reject” to D. In a case of “accept”, RS will keep the D’s credential for future communication.

This protocol is essential for P to be able to access the service provided by D, since the future service to P requires RS to act as the hub to bridge P and D.

2. Remote consultation phase.

In this phase, the patient P communicates with the doctor D by sending Ticket\(_{p,d}\) to D. In turn, the doctor D replies to the patient P to confirm the establishment of the communication channel. The communication before P and D is conducted via the home server HS and the remote server RS. Notice that we have omitted the ticket granting phase for P as it is same as that in Scenario 2. Therefore, similar to Scenario 2, the subliminal identity of P can still be used to protect the patient. This requires a routing table (6.4) which contains the information of P and D including their IDs, patient’s subliminal ID, home domain and current location. We describe it later in this chapter.

(a) P \(\rightarrow\) (RS) \(\rightarrow\) D : P\(^*_1\), D, \(n_p\), Ticket, MAC\(_{k_s}\)(P\(^*_1\), D, \(n_p\), Ticket)

(b) D \(\rightarrow\) (RS) \(\rightarrow\) P : D, P\(^*_1\), \(n_p\), ok, MAC\(_{k_s}\)(D, P\(^*_1\), \(n_p\), ok)

(c) P \(\rightarrow\) (RS) \(\rightarrow\) D : \(\cdots\) Consultation request
This communication channel can last while there is a necessity. All the communication messages are encrypted with session key $k_s$, which provides the secure communication channel.

### 6.3.4 Scenario 4

In this scenario (Fig. 6.5), suppose a patient $P$ lives in a remote rural area and has registered with her home domain server $RS$. A doctor $D$ resides in a major city and has also registered with his home domain server $HS$. $P$ needs to travel to different places temporarily in which her communication requests are being forwarded to $RS$.

![Figure 6.5: Scenario 4. The figure on the left-hand side shows the case while both $P$ and $D$ have a separate home server and both reside in their own home. The figure on the right-hand side shows the consultation phase.](image)

$P$, when is in her remote area, sends requests to her home server $RS$ in order to establish remote consultation with $D$. The $RS$ in turn contacts $HS$ in order to obtain a service ticket for $P$. A secure communication session can be established through home servers $RS$ and $HS$. The $P$’s home server $RS$ is trusted by $D$’s home server $HS$ in which her requests are being sent to. Since $RS$ can authenticate $P$, also $HS$ and $RS$ are trusted each other, the authentication of $P$ by $HS$ is based on a mutual trust.

Once the authentication is done, the $HS$ will issue a service ticket and send it to $RS$ who can then forward it to $P$. Since $D$ is located at his home domain and already registered, he does not need to be authenticated again.

1. Obtaining a ticket.
In this phase, the patient $P$ needs to contact its home server $RS$ to obtain a service ticket for a secure communication channel with the doctor who resides in $HS$. The $RS$ will contact the $HS$ in order to obtain a service ticket for $P$.

(a) $P \to RS : P^s_1, RS, D, HS, n_p, MAC_{p,rs}(P^s_1, RS, D, HS, n_p)$

The patient $P$ contacts its home server $RS$ as the first step to provide the information of the doctor’s ID and its home domain, along with a nonce $n_p$. The $RS$ authenticates $P$ by verifying the MAC.

(b) $RS \to HS : RS, HS, [P, P^s_1, D]_{K_{hs,rs}}, n_p, n_{rs}, MAC_{K_{hs,rs}}(RS, HS, p, P^s_1, D, n_p, n_{hs})$

In this step, the patient’s home server $RS$ contacts the home server of $D$ to request a service ticket for $P$ to communicate with $D$.

(c) $HS \to RS : HS, RS, Package, MAC_{K_{hs,rs}}(HS, RS, Package)$

where

$$\text{Package} = [P^s_1, D, k_s, \text{Ticket}]_{K_{hs,rs}}$$

$$\text{Ticket} = [P, P^s_1, D, k_s, T]_{K_{d,hs}}$$

$\text{Package}$ is sent to $RS$, so that it can forward it to $P$ in the next step. Here, $T$ is a timestamp. The $\text{Package}$ contains $\text{Ticket}$ which is sent to $P$ in the next step. In order for $P$ and $D$ to establish a secure channel for the consultation, a session key $k_s$ is embedded in $\text{Package}$, to ensure both $P$ and $D$ will obtain it. The doctor $D$ will obtain $k_s$ from the service ticket $\text{Ticket}$.

(d) $RS \to P : RS, P^s_1, [P^s_2]_{K_{p,rs}}, n_p, \text{Package}, MAC_{K_{p,rs}}(RS, P^s_1, P^s_2, n_p, \text{Package})$

In this step, $\text{Package}$ is delivered to $P$ by $RS$. $P$ therefore obtains the service ticket $\text{Ticket}$ and the secret session key $k_s$; hence $P$ is ready to contact his doctor $D$ for consultation. A new subliminal ID $P^s_2$ for patient $P$ is delivered to $P$ in this step. As a necessary matter, $P$ needs to check if $n_p$ is the same as the one initialised by itself in the first step. If the checking step returns true, the entire protocol run is complete.

2. Remote consultation phase.

In this phase, the patient $P$ communicates with the doctor $D$ by sending $\text{Ticket}_{p,d}$ to $D$. In turn, the doctor $D$ replies to the patient $P$ to confirm the establishment of the communication channel. The communication before $P$ and $D$ is conducted via the home server $HS$ and the remote server $RS$. 
6.4 Routing Tables

The routing tables play an important role in the protocols. Both HS and RS need to maintain a routing table, respectively. The table should contain the information of the real IDs of users for both patients and doctors, the up-to-date subliminal IDs for patients, home servers of users and the current server. This is updated once a user is registered, a change of subliminal ID, and change of user location to a different domain. In Table 6.3, we provide an example for some cases of our protocol.

<table>
<thead>
<tr>
<th>ID</th>
<th>Sub ID</th>
<th>Home Server</th>
<th>Current Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>P1₁ᵣ</td>
<td>RS</td>
<td>RS</td>
</tr>
<tr>
<td>P2</td>
<td>P2ᵢⱼ</td>
<td>RS</td>
<td>HS</td>
</tr>
<tr>
<td>D1</td>
<td></td>
<td>HS</td>
<td>HS</td>
</tr>
<tr>
<td>D2</td>
<td></td>
<td>HS</td>
<td>RS</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

6.5 Clinical Examples

In this section, we present two clinical examples for our protocols. We describe how the service should be delivered and how patients and doctors should be registered and managed in our system to capture the features of user mobility and distributed service.
6.5. Clinical Examples

For a distributed telemedicine system, using multiple servers to manage the services is essential to make the system efficient and practical for telemedical services. Therefore, we set up multiple servers in terms of geographical locations. We use the term "telemedicine servers" to name such service. In our protocols, telemedicine servers are also authentication servers for our security services. Therefore, the telemedicine servers are located at various geographical locations and also play the role of the authentication server. In order to use the service, all doctors and patients have to register with their local telemedicine server. There are the following entities in our examples.

- Telemedicine servers: The servers which manage doctors and patients to provide authentication service and secure channels for communication.
- Doctors: They are medical specialists who provide special service to patients. They usually reside in large medical centres such as hospitals or specialised clinics. A doctor has registered with a telemedicine server at his home location for example. He can, however, travel to a different location while can still provide his service to his patients.
- Patients: Patients are people who need to consult with a doctor on their illnesses. Patients have registered with a telemedicine server which is usually located at their home location. Patients might be located in rural and deprived areas and cannot access specialists easily. Any patient can travel to a different location and still obtain the requested consultation with a doctor.

Our protocols presented in this work capture the aforementioned application scenarios. The examples provided here elaborate two medical cases in order to sketch a comprehensive picture of our solutions’ applicability.

Our protocols allow flexible communication networks such as 4G LTE mobile service, wired internet service and WiFi networks. Without loss of generality, we assume that the telemedicine servers are web-based servers, where the users have registered with their valid ID.

Example 1. In this example, we show that how a patient who lives in a rural area does not have to travel to a metropolitan city or a facilitated hospital in order to receive a medical consultation with a specialist. Technically, this example can be explained by Scenario 4, which has been presented earlier. Suppose that a doctor Donald who has registered with the telemedicine server - New York City Telemedicine Service,
and a patient Petty who has registered with another telemedicine server - Kalawao County Telemedicine Service. Petty can visit a local clinic for her ordinary check-ups or consultations with a General Practitioner (GP); however, in chronic conditions or operations, her local GP cannot provide the professional advice. Then, GP refers Petty to visit the specialist, Donald. Therefore, she can request secure remote consultation to Donald, who is located in New York City.

With our protocol presented in Scenario 4, we can allow Petty and Donald to establish a secure and authenticated channel for an online consultation meeting. Petty only needs to contact the Kalawao County Telemedicine Service by the provided web service via a wired or wireless communication network. With the aid of the New York City Telemedicine Service, the Kalawao County Telemedicine Service can obtain a service ticket for Petty to consult with Doctor Donald over a secure and authenticated channel.

**Example 2.** Now, we assume that Doctor Donald is currently treating a patient Penny due to her chronic medical condition. Suppose both Penny and Donald have registered with the New York City Telemedicine Service; therefore, it is convenient for Penny to visit Donald in order to obtain his advice or treatment. However, as part of Penny’s job, she has to travel to a different city Los Angeles. Our Scenario 2 captures this case perfectly. The procedure is as follows.

Penny needs to contact the Los Angeles Telemedicine Service through its web service. According to our Scenario 2, even Penny has not registered with the Los Angeles Telemedicine Service, she can successfully receive a service ticket which allows her to establish a secure and authenticated remote communication channel with Donald. Therefore, Penny can still consult her Doctor Donald while she resides outside her registered region.

By employing our solutions, we ensure that a patient who is in a remote area can still benefit from secure remote consultation meetings with a specialist via remote servers which can significantly reduce the time and the cost involved in travelling long distances for medical appointments. We might highlight the advantage of distributed feature of our solutions. Any server in the system can serve as the point of attachment for patients and doctors; therefore, it solves the bottleneck problem in the centralised schemes.
6.6 Security Analysis

Under the aforementioned security assumptions our protocol is secure as following theorems:

We have defined earlier that we consider outside adversaries only. The outsiders are those people who are not registered with the system and the insiders who have been registered with their home server but are involved in the present protocol execution (i.e., they do not hold the secret keys which have been used in the current protocol run). All adversaries can launch active attacks. For simplicity, we denote by $\mathcal{A}$ the adversary.

**Theorem 6.1** The proposed protocol ensures the data confidentiality against the adversary $\mathcal{A}$.

A legitimate user shares a long-term symmetric key with its home server. Since the data transmitted between a user and its home server is encrypted with the long-term key in the ticket granting phase, to access the data, one must have the key. The adversary $\mathcal{A}$ can be a registered user (patient or doctor) and hold its own shared key with its home server, but it does not have others’ keys. Therefore, it is impossible for $\mathcal{A}$ to decrypt data if the encryption scheme is secure.

For each session, there is a new session key issued for patient-doctor communication and all data flows are encrypted with this key; therefore, it is secure against $\mathcal{A}$. The confidentiality while a patient is in a remote area is achieved with the data encrypted under a temporary key issued by the patient. Hence, $\mathcal{A}$ does not hold this key.

The security is also due to the assumption that the home server is trusted to all its users (patients and doctors). This ensures that the data received by the home server will not be revealed to $\mathcal{A}$. Also, the final session key $k_s$ which is selected by the home server and transmitted with the help of the home server (or remote server) cannot be obtained by the $\mathcal{A}$, as we have assumed that all underlying encryption algorithms are secure.

**Theorem 6.2** The proposed protocol ensures the data integrity against the adversary $\mathcal{A}$.

The data integrity is obtained with the Message Authentication Code (MAC) which is based on a secure cryptographic hash function, such as SHA-2. All transmitted
data are embedded in MAC with a MAC key, which is, by default, the long-term key shared by a user and its home server. This ensures that only those parties who have that key can compute and verify a MAC value. If MAC is secure, therefore, the integrity of the data transmission in our system is ensured.

Theorem 6.3 *The proposed protocols ensure the patient anonymity against the adversary $A$.*

The user anonymity is achieved with the subliminal ID of the user. In our protocols, even the $A$ who has its own key cannot know other users’ identities, though they can obtain the identity of their communication partner. The reason is threefold: (1) A subliminal identity is used only for one round of the protocol and a new subliminal ID is transmitted to the user securely. (2) The new subliminal ID is encrypted by the home server, therefore, only the corresponding user can decrypt it. (3) Our protocols also offer patient untraceability, which means that a patient cannot be traced back with its previous communication transcripts. The reason is that the subliminal ID is updated when a communication session is completed. The new session uses a new subliminal ID.

Theorem 6.4 *The proposed protocols provide the freshness in communication against any replay attack.*

We utilise nonces in our protocols. The following facts support our claim. The communication session between two parties is accompanied with a new and random nonce and the nonce must be returned to the initiator of the nonce to ensure the completion of the session and ensure that the freshness can be verified. Since the different communication session has a different nonce, the information from the previous session cannot be applied to the current session as it has a different nonce. In addition, the adversary can not use an old communication nonce in the future protocols; because the server keeps a table of all the nonces for a period of time. The freshness can be further enhanced by adding a timestamp $T$. Since it is a straightforward process, we omit it in this chapter.

Theorem 6.5 *The proposed protocols provide the authentication service.*

Since MACs are used in our protocols, our protocols provide symmetric-key based authentication service. That is, for two parties who share a symmetric key can
authenticate each other due to the fact that only the party who holds the key can compute the MAC value, therefore, the corresponding receiver of MAC value can verify it by using the same key.

The patient is granted a ticket, which is encrypted under the doctor’s key shared with its home server. If the doctor can decrypt it successfully, the doctor can be ensured that the ticket was created by its home server, which implicitly authenticates its home server. Also, if the doctor can successfully respond to the patient upon receiving the ticket, the patient then knows that the doctor is legitimate. Since the ticket contains the patient’s ID, the doctor will know that the patient is legitimate.

6.7 Conclusion

In this chapter, we proposed several security protocols for practical telemedicine applications. Our protocols offer security properties of data confidentiality, patient anonymity (untraceability), data integrity, data freshness and mutual authentication. Our protocols are featured with the capability of handling user mobility along with patient anonymity, which is not achieved in the previous studies. We presented four practical telemedicine scenarios, in which we showed that our protocols have provided all properties we predefined in our work. Our protocols are efficient in terms of computation cost and speed, as they require the symmetric-key cryptographic schemes only.
Chapter 7

Provably Secure Homomorphic Sign-encryption and Its Application in Patient Monitoring

Computerised patient monitoring systems have been widely adopted in healthcare applications in recent years. Although such systems have brought great efficiency and convenience to patients and health care givers, protecting users’ privacy and data security are still challenging issues. The main reason is that collected data by monitoring devices is vulnerable to various attacks. Therefore, such data can be protected via encryption schemes, which offer confidential protection to patient information. Although some systems adopt digital signature schemes on the top of encryption to ensure the authenticity of the information, handling such systems require additional computational resources and cryptographic key management. In this chapter, we propose a novel scheme offering lightweight computations which ensures data integrity, confidentiality and authentication. In addition, our scheme offers the following two features: (1) Signing and encrypting are carried out in one go, unlike the traditional encryption and signature schemes which are computed separately. (2) We allow the collected encrypted and signed data pieces to be aggregated without requiring decryption. The second feature confirms the significance of the first feature in that the traditional signcryption cannot be applied due to lacking of the homomorphic property. Our scheme is the first provably secure solution to the problem. We apply our scheme to patient monitoring devices to enhance computational efficiency, confidentiality and authentication.

7.1 Introduction

Owing to rapid growth in mobile health applications, healthcare providers offer various remote medical services to improve the efficiency and convenience to healthcare
services. These services also have great impacts on the improvement of individuals’ well-being, reducing the costs of hospitalisation and treatment [KGKW16].

In order to benefit from mobile healthcare systems, wireless body sensor networks, implantable medical devices, wearable mobile devices and mobile phones have been utilised to monitor patients health conditions, such as body temperature, blood pressure, heart rates, etc. The collected data are sent to a doctor and are useful for the doctor to know the patient’s medical condition in case of emergencies or the evaluation of an ongoing treatment, in a timely manner. This technology is very helpful for patients who reside in remote or deprived areas and require an ongoing medical care.

Encryption schemes have been commonly adapted to assure security and privacy in patient monitoring devices. Encryption techniques can help to protect patient information against disclosure to unauthorised parties. To achieve data authentication, a signature scheme is usually required in addition to encryption. Usually, the data is digitally signed and then encrypted. Therefore, additional computational resource is needed to adopt both encryption and signature schemes. However, those schemes might be inefficient in implementing in wearable devices due to the limited computation and communication capacity. There exists a technique “signcryption” which integrates signing and encrypting processes into one; therefore improves the computational efficiency.

When the amount of data is large, for example, the patient information from repeated and regular measurements of a patient health condition such as body temperature, heart rates and/or blood pressure, health data aggregation services provide promising applications in patient monitoring devices to improve the data verification efficiency. These information include spatial data and temporal data. Spatial data is necessary for pharmaceutical research and production, and temporal data is important to monitor the health condition of users and timely feedbacks [HZL+16]. On the other hand, protecting patients’ data and preserving their privacy have become a common concern preventing patients from participating in mHealth systems. The remote patient monitoring also incurs a high volume of vital data transferring to the server and receiver for storage and processing, which is important to be protected [ZCDL15]. With the tool of homomorphic encryption, encrypted data pieces can be aggregated in terms of either additive operations or multiplicative operations. Although fully homomorphic encryption is not computationally efficient for a practical use, partial homomorphic encryption (either fully additive or fully multiplicative)
can be efficiently constructed. However, the homomorphic property will become infeasible if a signature is added to the encryption.

There is no existing proper homomorphic signcryption scheme in the literature. Notice that Zhang et al. [ZYL11] introduced a homomorphic signcryption scheme, but its security cannot be properly proved, since the simulator cannot simulate the entire homomorphic signcryption; instead, the simulation for the encryption part was carried out separately without considering the signature verification. Actually, if the verification is considered, the adversary can differentiate which challenge message is encrypted by the challenger; therefore, it will not be semantically secure as claimed in their paper. In fact, it is indeed a challenge to achieve a provably secure homomorphic signcryption scheme mathematically.

We deem that a partially homomorphic signcryption scheme is necessary for our system, where we want to achieve an efficient remote patient monitoring system which captures the following features: The pieces of information collected from a patient via his/her wearable device are sign-encrypted before sent to a computer server managed by the healthcare provider, who in turn aggregates them into a single piece with decrypting them. Because of the additive homomorphic property, the sum of the data pieces is computed under the encryption shield. The encrypted sum is then forwarded to the corresponding doctor who can obtain the average result of the measurements.

The original signcryption scheme by Zheng [Zhe97] utilises the symmetric-like encryption approach, which cannot adopt homomorphism in encryption. We move slightly away from original signcryption, by introducing a useful variant. To differentiate it from the original signcryption, we name it “homomorphic sign-encryption”. In this work, we propose a homomorphic sign-encryption scheme to serve our goal. Our scheme can integrate sign-encrypted health data collected from a patient’s wearable device without the need of decryption. This feature is desirable when we assume that the computer server is not fully trusted. The aggregated data is then sent to the corresponding doctor who can then decrypt the aggregated data and verify its authenticity. With this approach, the doctor does not need to handle data collection and aggregation, while can ensure the security of the transmitted data. In our scheme, we assume that the receiver, i.e., the corresponding doctor is trusted by the patients. This assumption is reasonable in the ehealth scenarios. With this assumption, we are able to prove the security of our scheme formally.
Our Contribution. The contribution of our work can be summarised as follows. We propose the first secure homomorphic sign-encryption scheme. Our scheme is provably secure. The security analysis demonstrates that our scheme achieves IND-CPA and Weak Unforgeability (WUF) security goals, under the Decisional Diffie-Hellman assumption and the Computational Diffie-Helman assumption, respectively. Our scheme improves the secure data collection from the wearable sensor devices in patient monitoring, in which such devices have limited computational power.

The remaining of the chapter is organised as follows. We describe the related work in Section 7.2. In Section 7.3, we present our system model and give the definitions of our scheme and security model. In Section 7.4, we present our scheme followed by the security proof of our scheme in Section 7.5. We conclude the chapter in Section 7.6.

7.2 Related work

In the literature, there are several studies dealing with patients’ data security and privacy in distributed healthcare services and cloud computing systems [SFZ10, YBG+16, YRL11, ZLDC15]. They presented cryptographic solutions in order to ensure the monitoring data protection while are being aggregated and shared with the healthcare provider. The proposed solutions have employed signcryption schemes which combine signature and encryption schemes together in order to ensure authenticity and confidentiality.

In some studies, homomorphic encryption is used to provide secure data integration in monitoring devices. In the homomorphic encryption, the encryptions of different messages can be combined to compute either additive or multiplicative operations without revealing their inputs. In a fully homomorphic encryption, both additive and multiplicative operations can be carried out. The homomorphic encryption has been a useful method for designing secure computation protocols. In a study proposed by Boneh et al. [BGN05], the homomorphic properties of the current homomorphic public key systems are improved, in which given two ciphertexts, anyone can compute both addition and multiplication. Although it is not yet a fully homomorphic encryption, it has many useful applications. The underlying security of the proposed scheme, named BGN, is based on a new hardness assumption named Subgroup Decision Problem. Unfortunately, there is no any practical homomorphic
7.2. Related work

Although fully homomorphic encryption is not yet ready for practical use, there have been several works, which utilise lattice-based fully homomorphic encryption to medical system applications. For example, in [KS15], a medical system application was proposed based on the Gentry’s homomorphic encryption scheme [Gen09] to ensure data privacy in the public cloud.

With a natural thinking of homomorphism in digital signatures, Johnson et al. [JMSW02] proposed a homomorphic signature scheme, which unfortunately cannot work with any form of homomorphic encryption. Chan and Li [CL06] also proposed a BGN authentication scheme to convey the commitments on a message in order to provide statistically hiding and computationally binding properties under the subgroup decision problem.

There are many works in the literature which explored the applications of partial homomorphic encryption schemes such as Paillier’s additive homomorphic encryption [Pai99] and ElGamal encryption [El 85]. For example, Yi et al. [YBG+16] proposed a scheme that applied multi data servers with employing the Paillier and ElGamal cryptosystems in order to offer statistical analysis and also preserve patient privacy for wireless medical sensor devices. Han et al. [HZL+16] in another study illustrated a privacy-preserving aggregation scheme to support fault tolerance in order to aggregate health data in the cloud server. They also used BGN cryptosystem by Boneh et al. [BGN05], and proposed an aggregation protocol to compute the average.

The signcryption scheme by Zheng [Zhe97] is able to reduce the computational overhead of signature and encryption computation by combining them into a single algorithm. There are enormous applications of signcryption schemes which have been found in the literature. For example, in the studies proposed by Rao [Rao17] and Liu et al. [LHL15], attribute based signcryption schemes for secure sharing of health records and ensuring confidentiality and authenticity have been presented; whereas their schemes were not designed for body sensor networks which have limited computational complexity than cloud computing.

Despite of the usefulness of homomorphism in signcryption, it has not been explored thoroughly in research. As pointed out earlier, the homomorphic signcryption scheme due to Zhang et al. [ZYL11] cannot be properly proved. In this chapter, we will investigate and explore this field of research. Fortunately, we are able to construct a provably secure scheme for our specific application in the patient monitoring application.
7.3 Definitions and Models

7.3.1 System model

Our system consists of an honest-but-curious data server, a group of doctors and a group of patients. On top of these parties, there is a trusted server who sets up the entire system and is responsible for the management of cryptographic keys and user registration.

- **Patient**: A patient is equipped with a wireless wearable sensor device that collects specific his/her health data. Each measurement to the patient is sign-encrypted with our proposed cryptographic method and sent to the data server, who in turn aggregates these data without decryption and forwards it to the corresponding doctor.

- **Doctor**: In the case we considered in this chapter, a doctor is responsible for checking his patient’s health condition by the measurement data forwarded to him by the data server. The doctor possesses the decryption key to retrieve the aggregated data forwarded by the data server.

- **Data Server**: The data server is honest-but-curious, which means that the data server follows the correct procedure to aggregate the data items collected from patients and is interested in the patient information, while it does not launch any active attack. We consider only one data server in the system; however, our method can be naturally applied to a distributed environment for multiple data servers when the patient population is large. The data server can be located in different geographical locations.

Patient datasets might contain different types of data, such as blood pressure and body temperature. The evaluation of patient’s health condition is performed on one type of data, e.g. blood pressure. Our scheme can be performed on other types of data separately, namely, each type of data is sign-encrypted under the patient’s private key to ensure data authenticity and the corresponding doctor’s public key in order to ensure confidentiality. The sign-encrypted data can be verified and decrypted by the doctor. The data server, who collects these datasets, computes the sum for each type of sign-encrypted patient data. The data server does not hold the decryption key, hence cannot retrieve the patient data, while can help to aggregate the patient data.
7.3. Definitions and Models

We illustrate our system in Fig. 7.1, where a patient, his doctor and the data server are involved. Patient data \((m_0, \cdots, m_n)\) collected from patient’s wearable sensor are homomorphic-sign-encrypted as \((\text{HSE}(m_0), \cdots, \text{HSE}(m_n))\) and sent to the data server, who in turn computes the aggregated data items at a specific time. The integrated data \(\text{HSE}(m)\) is then sent to the doctor, who then decrypts and verifies the received message, computes the average of patient data on the patient’s health status (e.g. blood pressure) and evaluates the patient’s medical condition.

7.3.2 Complexity Assumptions

**Definition 7.1 (Computational Diffie-Hellman (CDH) Assumption)** Let \(G = \langle g \rangle\) be a cyclic group of prime order \(p\) generated by a generator \(g\). Given \(g, g^a, g^b \in G\) for randomly selected \(a, b \in \mathbb{Z}_p\), there exists an algorithm \(A\) that computes \(g^{ab}\) with the advantage

\[
\text{Adv}_{A}^{\text{CDH}} = \Pr \left[ g^{ab} \leftarrow A(G, p, g, g^a, g^b) \mid a, b \in \mathbb{Z}_p, G = \langle g \rangle \right].
\]  

(7.1)

The CDH assumption assumes that the advantage \(\text{Adv}_{A}^{\text{CDH}}\) is negligible for any probabilistic polynomial time (PPT) algorithm \(A\) under the security parameter \(1^\lambda\).

**Definition 7.2 (Decisional Diffie-Hellman (DDH) Assumption)** Let \(G = \langle g \rangle\) and \(a, b \in \mathbb{Z}_p\) as described in the CDH assumption. Given \(g, g^a, g^b\), there exists an
7.3. Definitions and Models

algorithm $A$ that distinguishes $g^{ab}$ with a random element $Z \in_R \mathbb{G}$ with the advantage

$$\text{Adv}_A^{\text{DDH}} = |\Pr[1 \leftarrow A(\mathbb{G}, p, g, g^a, g^b, g^{ab})] - \Pr[1 \leftarrow A(\mathbb{G}, p, g, g^a, g^b, Z)]| \in_R \mathbb{G}|$$

(7.2)

The DDH assumption assumes that the advantage $\text{Adv}_A^{\text{DDH}}$ is negligible for any PPT algorithm $A$ under the security parameter $1^\lambda$.

### 7.3.3 The Definition of Homomorphic Sign-encryption Scheme

**Definition 7.3 (Homomorphic Sign-encryption) A** homomorphic sign-encryption (HSE) scheme consists of the following five algorithms:

- **params $\leftarrow \text{Setup}(1^\lambda)$**. Taking as input a security parameter $1^\lambda$, it outputs the system public parameters $\text{params}$.

- **$(pk_s, sk_s) \leftarrow \text{KeyGen}_s(\text{params})$**. Taking as input the system public parameters $\text{params}$, it outputs a pair of public key $pk_s$ and secret key $sk_s$ of a sender (patient).

- **$(pk_r, sk_r) \leftarrow \text{KeyGen}_r(\text{params})$**. Taking as input the system public parameters $\text{params}$, it outputs a pair of public key $pk_r$ and secret key $sk_r$ of a receiver (doctor).

- **$\text{HSE}(m) \leftarrow \text{Sign-encrypt}(\text{params}, pk_r, sk_s, m)$**. Taking as input public parameters $\text{params}$, public key $pk_r$ of doctor, private key $sk_s$ of the patient, and a plaintext message $m$ in the message space $M$, it outputs a homomorphic sign-encryption $\text{HSE}(m)$.

- **$m \leftarrow \text{De-sign-encrypt}(\text{params}, pk_s, sk_r, \text{HSE}(m))$**. Taking as input public parameters $\text{params}$, a public key $pk_s$ of the patient, a private key $sk_r$ of doctor, and a ciphertext $\text{HSE}(m)$, it outputs plaintext message $m$.

- **$0/1 \leftarrow \text{Verify}(\text{params}, pk_s, sk_r, \text{HSE}(m), m')$**. Taking as input public parameters $\text{params}$, a public key $pk_s$ of the patient, a private key $sk_r$ of doctor, a ciphertext $\text{HSE}(m)$, and a message $m'$, it outputs 1, if $m = m'$; otherwise it outputs 0.

Remark that the system public parameters $\text{params}$ is omitted if it is clear in the context. An HSE scheme is required to have ciphertext homomorphism as the following algorithm.
7.3. Definitions and Models

- \( \text{HSE}(m) \leftarrow \text{IntSign-encrypt}(\text{HSE}(m_1), \ldots, \text{HSE}(m_n)) \). Taking as input \( \text{HSE}(m_1), \ldots, \text{HSE}(m_n) \), it outputs the integrated homomorphic sign-encryption \( \text{HSE}(m) \), where \( m = m_1 + \cdots + m_n \).

**Definition 7.4 (Completeness)** An HSE scheme is complete if the following statement is always true.

\[ \forall m \in M, \text{params} \leftarrow \text{Setup}(1^\lambda), \quad (pk_s, sk_s) \leftarrow \text{KeyGen}_s(\text{params}), \]
\[ (pk_r, sk_r) \leftarrow \text{KeyGen}_r(\text{params}), \quad \text{HSE}(m) \leftarrow \text{Sign-encrypt}(\text{params}, pk_r, sk_r, m), \]
\[ 1 \leftarrow \text{Verify}(\text{params}, pk_s, sk_r, \text{HSE}(m), \text{De-sign-encrypt}(\text{params}, pk_s, sk_r, \text{HSE}(m))). \]

### 7.3.4 Security model

**Definition 7.5 (Confidentiality)** An HSE scheme is semantically secure against chosen plaintext attacks (IND-CPA) if no PPT adversary \( \mathcal{A} \) wins the following game with non-negligible advantage with the security parameter \( \lambda \).

1. **Setup Phase.** The simulator \( \mathcal{S} \) runs \( \text{Setup} \) to obtain system public parameters \( \text{params} \). Then, the simulator \( \mathcal{S} \) runs the key generation algorithm \( \text{KeyGen}_r \) to obtain a public key and private key pair \( (pk_r, sk_r) \) for the receiver, it gives \( (pk_r, \text{params}) \) to the adversary \( \mathcal{A} \).

2. **Challenge Phase.** \( \mathcal{A} \) generates two plaintexts \( m_0, m_1 \in M \) and a private key \( sk_s \) of the sender, and sends to \( \mathcal{S} \). Then, \( \mathcal{S} \) sets \( \text{HSE}(m_b) \leftarrow \text{Sign-encrypt}(\text{params}, pk_r, sk_s, m_b) \) for a random bit \( b \leftarrow \{0, 1\} \). It sends \( \text{HSE}(m_b) \) to \( \mathcal{A} \).

3. **Guess Phase.** At the end of the game, \( \mathcal{A} \) outputs a bit \( b' \in \{0, 1\} \) to \( \mathcal{S} \) and wins the game if \( b' = b \).

The adversary \( \mathcal{A} \)'s advantage in the above game is defined as

\[ \text{Adv}_{\mathcal{A}}^{\text{IND-CPA}} = \left| \Pr[b' = b] - \frac{1}{2} \right|. \]

**Definition 7.6 (Unforgeability)** An HSE scheme is weakly unforgeable if no PPT forger \( \mathcal{F} \) has a non-negligible advantage in the following game:

1. **Setup Phase.** The simulator \( \mathcal{B} \) runs \( \text{Setup}, \text{KeyGen}_s \) and \( \text{KeyGen}_r \) to obtain two pairs of public key and private key \( (pk_s, sk_s) \) and \( (pk_r, sk_r) \). \( \mathcal{B} \) gives \( (pk_s, pk_r, \text{params}) \) to forger \( \mathcal{F} \).
2. Forgery Phase. Finally, $F$ returns a valid signature $HSE(m^*)$.

The forger $F$'s advantage in the above game is defined as

$$\text{Adv}_F^{WUF} = \Pr \left[ 1 \leftarrow \text{Verify} \left( \text{params, pk}_s, sk_r, HSE(m^*), \text{De-sign-encrypt}(\text{params, pk}_s, sk_r, HSE(m^*)) \right) \right].$$

Note that the forger $F$ is not allowed to perform $\text{Sign-encrypt}$ queries to simulator $B$.

7.4 Our Proposed Scheme

In this section, we propose our homomorphic sign-encryption scheme. An HSE scheme consists of the following algorithms.

- $\text{params} \leftarrow \text{Setup}(1^\lambda)$. Taking as input the security parameter $1^\lambda$, it outputs system parameters $\text{params} = (p, g)$ where $\mathbb{G} = \langle g \rangle$ is a group of prime order $p$, generated by a generator $g$.

- $(pk_s, sk_s) \leftarrow \text{KeyGen}_s(\text{params})$. Taking as input $\text{params}$, the algorithm randomly selects a private key $sk_s = w \in \mathbb{Z}_p$, and computes the corresponding public key $pk_s = h = g^w$ for the sender (patient).

- $(pk_r, sk_r) \leftarrow \text{KeyGen}_r(\text{params})$. Taking as input $\text{params}$, the algorithm randomly selects a private key $sk_r = (x_0, x_1, x_2) \in \mathbb{Z}_p^3$, and computes the corresponding public key $pk_r = (y_0, y_1, y_2) = (g^{x_0}, g^{x_1}, g^{x_2})$ for the receiver (doctor).

- $HSE(m) \leftarrow \text{Sign-encrypt}(pk_r, sk_s, m)$. Taking as input public key $pk_r$ of the receiver, secret key $sk_s$ of the sender, and a plaintext message $m \in \mathbb{M} = \{0, 1\}^l$ for $l \leq n$ where $n = 32$, it computes

$$C_0 = g^t, \quad C_1 = g^m y_0^t, \quad C_2 = y_1^m y_2^t,$$

where $t \in \mathbb{Z}_p$. It outputs $HSE(m) = (C_0, C_1, C_2)$ as the homomorphic sign-encryption.

- $m \leftarrow \text{De-sign-encrypt}(\text{params, pk}_s, sk_r, HSE(m))$. Given a homomorphic sign-encrypted message $HSE(m)$, the public key $pk_s = h$ of the sender, the private key $sk_r = (x_0, x_1, x_2)$ of the receiver, the message is computed by

$$m' = \log_g \frac{C_1}{C_0}.$$
7.5. Security Analysis

Then the algorithm runs the below Verify algorithm to verify the message $m'$.
If it outputs 1, the message $m'$ is accepted, and the algorithm outputs $m = m'$.
Otherwise, the algorithm outputs $\bot$, which is an abort symbol.
The correctness can be verified as

$$m = \log g \frac{C_1}{C_0} = \log g \frac{y_0^{m_1}}{g^{t_0}} = \log g \frac{y_0^{m_1} g^{t_0}}{g^{t_0}} = \log g^m = m.$$  

- $0/1 \leftarrow \text{Verify}(\text{params}, pks, sk_r, \text{HSE}(m), m')$. The verification algorithm outputs 1 if

$$C_2 = h^{x_1 m'} C_0^{x_2}.$$  

Otherwise, the sign-encryption is rejected and it outputs 0.
The correctness can be verified as

$$C_2 = h^{x_1 m} C_0^{x_2} = g^{w x_1 m} g^{t_2} = y_1^{w x_1} y_2.$$  

- $\text{HSE}(m) \leftarrow \text{IntSign-encrypt}(\text{HSE}(m_1), \ldots, \text{HSE}(m_n))$. The algorithm parses the sign-encryption $\text{HSE}(m_i)$ as $(C_{i,0}, C_{i,1}, C_{i,2})$ with randomness $t_i$. The algorithm integrates the sign-encryption by calculating

$$C_0 = \prod_{i=1}^n C_{i,0} = \prod_{i=1}^n g^{t_i} = g^{\sum_{i=1}^n t_i},$$

$$C_1 = \prod_{i=1}^n C_{i,1} = \prod_{i=1}^n g^{m_i y_0^{t_i}} = g^{\sum_{i=1}^n m_i y_0^{t_i}},$$

$$C_2 = \prod_{i=1}^n C_{i,2} = \prod_{i=1}^n y_1^{w m_i} y_2^{t_i} = y_1^{w \sum_{i=1}^n m_i} y_2^{\sum_{i=1}^n t_i}.$$  

Taking $m = \sum_{i=1}^n m_i$ and $t = \sum_{i=1}^n t_i$, the integrated sign-encryption $\text{HSE}(m) = (C_0, C_1, C_2)$ has the same form of the original sign-encryption. Finally, the algorithm outputs $\text{HSE}(m)$.

7.5 Security Analysis

Theorem 7.1 If there exists a PPT algorithm $A$ that can break the IND-CPA security of the HSE scheme with advantage $\text{Adv}^\text{IND-CPA}_A$, then there exists a PPT algorithm $B$ that can solve the Decisional Diffie-Hellman (DDH) problem with advantage

$$\text{Adv}^\text{DDH}_B \geq \frac{\text{Adv}^\text{IND-CPA}_A}{2}.$$
7.5. Security Analysis

**Proof:** Suppose a PPT algorithm $S$ that acts as the simulator of the system. We present a series of games (Game 0, Game 1 and Game 2) as follows.

- **Game 0.** This is the original IND-CPA game for our HSE scheme.
  1. The simulator $S$ runs $\text{Setup}$ to obtain system public parameters $\text{params} = (p, g)$. Then $S$ runs $\text{KeyGen}_r$ to generate a receiver key pair $(sk_r, pk_r) = ((x_0, x_1, x_2), (y_0, y_1, y_2))$, and passes $(pk_r, \text{params})$ to the adversary $A$.
  2. The adversary $A$ generates two plaintexts $m_1, m_2 \in M$ and a sender private key $sk_s = w$. The simulator computes the sign-encryption $\text{HSE}(m_b) = (C_0, C_1, C_2)$ normally with a random bit $b \leftarrow \{0, 1\}$ where
     \[
     t \in \mathbb{Z}_p, \quad C_0 = g^t, \quad C_1 = g^{m_w} y_0^t, \quad C_2 = y_1^{w_m} y_2^t.
     \]
     The simulator $S$ sends $\text{HSE}(m_b)$ to the adversary $A$.
  3. Finally, $A$ outputs a bit $b' \in \{0, 1\}$. If $b = b'$, $A$ wins the game and $S$ outputs 1. Otherwise $S$ outputs 0.

- **Game 1.** This game is the same as **Game 0** except that the simulator replaces $y_0^t$ with a random element $R_0 \in G$ in computing $C_1$ in the step 2 as
     \[
     t \in \mathbb{Z}_p, \quad R_0 \in \mathbb{G}, \quad C_0 = g^t, \quad C_1 = g^{m_w} [R_0], \quad C_2 = y_1^{w_m} y_2^t.
     \]

- **Game 2.** This game is the same as **Game 1** except that the simulator replaces $y_2^t$ with a random element $R_1 \in G$ in computing $C_2$ in the step 2 as
     \[
     t \in \mathbb{Z}_p, \quad R_0, R_1 \in \mathbb{G}, \quad C_0 = g^t, \quad C_1 = g^{m_w} R_0, \quad C_2 = y_1^{w_m} [R_1].
     \]

In the following, we analyse the three games presented above under the DDH assumption. Then, we construct a distinguisher algorithm $B$ and estimate its probability in distinguishing differences among games. Let $E_i$ be the event that $A$ wins the **Game $i$** (i.e. $1 \leftarrow S$) for $i = 1, 2, 3$. By Definition 7.5, the advantage of $A$ winning the original game (**Game 0**) is

\[
\text{Adv}_{A}^{\text{IND-CPA}} = \left| \Pr[E_0] - \frac{1}{2} \right|. \quad (7.3)
\]

**Lemma 7.2** If an adversary $A$ can distinguish the difference between **Game 0** and **Game 1**, an algorithm $B$ can be constructed to solve a DDH problem with the advantage

\[
\text{Adv}_{B}^{\text{DDH}} = \left| \Pr[E_0] - \Pr[E_1] \right|. \quad (7.4)
\]
Proof: The algorithm $B$ obtains a DDH instance $(p, g, g^a, g^b, Z)$ from its challenger. The algorithm $B$ proceeds the following game with the adversary $A$ for our HSE scheme.

1. The algorithm $B$ samples $x_1, x_2 \in \mathbb{Z}_p$, and computes
   \[ y_0 = g^b, \quad y_1 = g^{x_1}, \quad y_2 = g^{x_2}. \]
   Then, $B$ packs $pk_r = (y_0, y_1, y_2)$ and $\text{params} = (p, g)$, and sends them to the adversary $A$.

2. The adversary $A$ generates two plaintexts $m_1, m_2 \in M$ and a sender private key $sk_s = w$. The algorithm $B$ computes the sign-encryption $\text{HSE}(m_b) = (C_0, C_1, C_2)$ with a random bit $b \leftarrow \{0, 1\}$ where
   \[ C_0 = g^a, \quad C_1 = g^{m_b Z}, \quad C_2 = y_1^{wmb}(g^a)^x_2. \]
   Then, $B$ sends $\text{HSE}(m_b)$ to the adversary $A$.

3. Finally, $A$ outputs a bit $b' \in \{0, 1\}$. If $b = b'$, $A$ wins the game and $B$ outputs 1. Otherwise $B$ outputs 0.

If $Z = g^{ab}$, the above game is exactly the same as the Game 0. Thus, we have
\[
\text{Pr}[1 \leftarrow B \mid Z = g^{ab}] = \text{Pr}[E_0]. \tag{7.5}
\]
Otherwise, $Z \in \mathbb{Z}_p \mathbb{G}$ is a random element in $\mathbb{G}$, and the above game is exactly the same as the Game 1. Thus, we have
\[
\text{Pr}[1 \leftarrow B \mid Z \in \mathbb{Z}_p \mathbb{G}] = \text{Pr}[E_1]. \tag{7.6}
\]
Therefore, by combining Eqs. (7.2), (7.5), and (7.6), we directly have Eq. (7.4) and complete the proof of this lemma.

Lemma 7.3 If an adversary $A$ can distinguish the difference between Game 1 and Game 2, an algorithm $B$ can be constructed to solve a DDH problem with the advantage
\[
\text{Adv}^\text{DDH}_B = |\text{Pr}[E_1] - \text{Pr}[E_2]|. \tag{7.7}
\]
Proof: The algorithm $B$ obtains a DDH instance $(p, g, g^a, g^b, Z)$ from its challenger. The algorithm $B$ proceeds the following game with the adversary $A$ for our HSE scheme.
7.5. Security Analysis

1. The algorithm $B$ samples $x_0, x_1 \in R \mathbb{Z}_p$, and computes

$$y_0 = g^{x_0}, \quad y_1 = g^{x_1}, \quad y_2 = g^b.$$ 

Then, $B$ packs $pk_r = (y_0, y_1, y_2)$ and $\text{params} = (p, g)$, and sends them to the adversary $A$.

2. The adversary $A$ generates two plaintexts $m_1, m_2 \in M$ and a sender private key $sk_s = w$. The algorithm $B$ computes the sign-encryption $\text{HSE}(m_b) = (C_0, C_1, C_2)$ with a random bit $b \leftarrow \{0, 1\}$ where

$$R_0 \in R \mathbb{G}, \quad C_0 = g^a, \quad C_1 = g^{m_b} R_0, \quad C_2 = y_1^{w m_b} Z.$$ 

Then, $B$ sends $\text{HSE}(m_b)$ to the adversary $A$.

3. Finally, $A$ outputs a bit $b' \in \{0, 1\}$. If $b = b'$, $A$ wins the game and $B$ outputs 1. Otherwise $B$ outputs 0.

If $Z = g^{ab}$, the above game is exactly the same as the Game 1. Thus, we have

$$\Pr \left[ 1 \leftarrow B \mid Z = g^{ab} \right] = \Pr[\mathcal{E}_1]. \quad (7.8)$$

Otherwise, $Z \in R \mathbb{G}$ is a random element in $\mathbb{G}$, and the above game is exactly the same as the Game 2. Thus, we have

$$\Pr \left[ 1 \leftarrow B \mid Z \in R \mathbb{G} \right] = \Pr[\mathcal{E}_2]. \quad (7.9)$$

Therefore, by combining Eqs. (7.2), (7.8), and (7.9), we directly have Eq. (7.7) and complete the proof of this lemma.

**Lemma 7.4** In the Game 2, the adversary $A$ has no advantage, i.e.

$$\Pr[\mathcal{E}_2] = \frac{1}{2}. \quad (7.10)$$

**Proof:** In Game 2, the adversary is given the sign-encryption $\text{HSE}(m_b) = (g^t, g^{m_b} R_0, y_1^{w m_b} R_1)$ where $R_0$ and $R_1$ are independent random elements, which work as one-time pads, rendering the bit $b$ independent from adversary $A$’s view. Therefore, the adversary $A$ has no advantage of winning the game other than a random guess. By combining Eqs. (7.3), (7.4), (7.7), and (7.10), we obtain

$$\text{Adv}_A^{\text{IND-CPA}} \leq 2 \cdot \text{Adv}_B^{\text{DDH}}.$$ 

Thus it completes the proof. Since the DDH assumption states that $\text{Adv}_B^{\text{DDH}}$ is negligible, we have $\text{Adv}_A^{\text{IND-CPA}}$ is negligible for all PPT adversaries.
7.6 Conclusion

Theorem 7.5 If there exists a PPT algorithm $\mathcal{A}$ that can break the weak unforgeability of the HSE scheme with advantage $\text{Adv}^\text{WUF}_A$, then there exists a PPT algorithm $\mathcal{B}$ that can solve the Computational Diffie-Hellman (CDH) problem with advantage $\text{Adv}^\text{CDH}_B \geq \text{Adv}^\text{WUF}_A$.

Proof: The algorithm $\mathcal{B}$ obtains a CDH instance $(p, g, g^a, g^b)$ from its challenger. The algorithm $\mathcal{B}$ simulates the weak unforgeability game (Definition 7.6) for the adversary $\mathcal{A}$.

1. The algorithm $\mathcal{B}$ samples $x_0, x_2 \in \mathbb{Z}_p$, and computes

   $$y_0 = g^{x_0}, \quad y_1 = g^b, \quad y_2 = g^{x_2}.$$  

   Then, $\mathcal{B}$ packs $pk_a = g^a$, $pk_r = (y_0, y_1, y_2)$, and $\text{params} = (p, g)$. After that, $\mathcal{B}$ sends them to the adversary $\mathcal{A}$.

2. Ultimately, $\mathcal{A}$ outputs a valid homomorphic sign-encryption $\text{HSE}(m^*) = (C_0^*, C_1^*, C_2^*)$ on arbitrary $m^* \in M$.

Finally, the algorithm $\mathcal{B}$ is able to compute $g^{ab}$ by

$$g^{ab} = \left( \frac{C_2^*}{C_0^{x_2}} \right) \left( \log_g \frac{C_1^*}{C_0^{x_0}} \right)^{-1}.$$  

Therefore, we immediately obtain the theorem. Since the CDH assumption states that $\text{Adv}^\text{CDH}_B$ is negligible, we have $\text{Adv}^\text{WUF}_A$ is negligible for all PPT adversaries.

7.6 Conclusion

Homomorphic signcryption is useful in many applications. However, it is a research challenge to accommodate the homomorphism feature in a traditional signcryption scheme. In this chapter, we proposed a variant of signcryption, named signcryption, which leads to a novel homomorphic sign-encryption scheme. With our proposed scheme, we gave an efficient approach for patient monitoring services. We formally proved the security of our proposed scheme.
Chapter 8

Conclusion and Future Work

We have presented the comprehensive research on the protection of health data privacy and security in Electronic Health Record Systems. The thesis captures several most interesting research themes and provides the solutions to the research problems we have identified. Our work is based on cryptographic and access control primitives, which served as building blocks for us to design secure and privacy enhanced EHRS.

We proposed and described an EHR system model which can provide scalability and flexibility of EHR systems in distributed environments while preserving privacy and security of EHR data. We investigated one of the solutions to the security and privacy issues on distributed EHR systems, which avoids using a single trusted authority. With ABE, we can provide fine-grained access control and flexible policies for secure EHR system infrastructures.

We presented an access control mechanism for an EHR system with the hybrid cloud structure, which allows us to handle various types of users who possess different access privileges. Our system features dynamic policy transformation based on some useful cryptographic building blocks. Our novel policy transformation approach enables EHR data to be transferred from a private cloud to a public cloud with the corresponding transformation in the access control policy. We proposed an implementation protocol for an application scenario.

We introduced the policy similarity to deal with multiple access control policies, which adds a layer of functionality to implementing the XACML system. A similarity value specifies the applicability of policies combination. Our approach provides a solution for the decision maker to decide whether two or more XACML policies should be combined, based on the value of policy similarity. Our access control policy combination solution has been shown useful in our case study in order to preserve the privacy of users and ensure data security.
We proposed several security protocols for practical telemedicine applications. Our protocols offer security properties of data confidentiality, patient anonymity (untraceability), data integrity, data freshness and mutual authentication. Our protocols are featured with the capability of handling user mobility along with patient anonymity, which is not achieved in the previous studies. Our protocols are efficient in terms of computation cost and speed, as they require the symmetric-key cryptographic schemes only.

Homomorphic signcryption is useful in many applications. However, it is a research challenge to accommodate the homomorphism feature in a traditional signcryption scheme. We proposed a variant of signcryption, named sign-encryption, which leads to a novel homomorphic sign-encryption scheme. With our proposed scheme, we gave an efficient approach for patient monitoring services. We formally proved the security of our proposed scheme.

Although we have provided useful solutions to the security problems we have identified in EHRS, there are still many security issues in EHRS. One of problems we have seen is that the authority who helped to set up the system has been regarded as a fully trusted (or semi-trusted) entity. This assumption is reasonable for most of applications. It would be desirable that we could reduce the trust level of the authority, while we can achieve the same level of security. We leave it to our future work.
Bibliography


[BDK+15] Antonia Bertolino, Said Daoudagh, Donia El Kateb, Christopher Henard, Yves Le Traon, Francesca Lonetti, Eda Marchetti, Tejeddie
BIBLIOGRAPHY


[DRK+13] Georg Duftschmid, Christoph Rinner, Michael Kohler, Gudrun Hübner-Blod, Samrend Saboor, and Elske Ammenwerth. The EHR-ARCHE project: Satisfying clinical information needs in a shared electronic


[WLL12a] Changji Wang, Xuan Liu, and Wentao Li. Implementing a personal health record cloud platform using ciphertext-policy attribute-based...


