uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

u Ottawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Fang Tian

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.A.Sc. (Electrical and Computer Engineering)

GRADE / DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Security Architecture and Its Implementation in a Multi Mobile Agent E-Hospital

TITRE DE LA THÈSE / TITLE OF THESIS

A. Carlisle

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

Liam Peyton

Anil Somayaji

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Security Architecture and Its Implementation in a Multi Mobile Agent E-Hospital

Fang Tian

Thesis submitted to the

Faculty of Graduate and Post Doctoral Studies

In partial fulfillment of the requirements

For the M.A.Sc in Electrical Engineering Degree

University of Ottawa

Your file Votre référence
ISBN: 978-0-494-74147-4
Our file Notre référence
ISBN: 978-0-494-74147-4

Abstract

This thesis describes the design and the implementation of a security architecture for an e-hospital. The e-hospital uses mobile agent technologies in its function implementation. The mobile agent technology has notable advantages over the traditional client/server technology. However, its advantages make it more vulnerable to threats. We use ISO17799, an international standard for Information security management, as our guide to approach the security problems in this e-hospital. We reach the security objectives for this e-hospital by risk assessments of the current system, compliance with the patients' privacy protection regulations and the consideration of end users' requirements. We divide the whole e-hospital into different domains according to their security levels and propose protections for each domain. We use identity-based two party key agreement protocols to address the network security in this e-hospital. Finally, we extend Bagga and Molva's policy-based cryptographic scheme to address the end user's access control and non-repudiation. Our proposed cryptographic scheme is based on the hardness of the Generalized Bilinear Diffie-Hellman Problem (GBDH problem) in groups. We implement our proposed cryptographic scheme in Java. The test results show that the performance of our proposed cryptographic scheme is acceptable using the elliptic curve defined over large prime fields.

Table of Contents

List of Tables and Figures

Legend

AES: Advanced Encryption Standard

CCA: Chosen Ciphertext Attack

CL-PKC: Certificateless Public Key Cryptography

ECC: Elliptic Curve Cryptography

ECDH: Elliptic Curve Diffie-Hellman protocol

ECDSA: Elliptic Curve Digital Signature Algorithm

ECIES: Elliptic Curve Integrated Encryption Scheme

ECMQV: Elliptic Curve Menezes-Qu-Vanstone protocol

IBE: Identity-Based Encryption

IDS: Intrusion Detection System

JADE: Java Agent Development Framework

PBC: Policy-Based Cryptography

PHIPA: Personal Health Information Protection Act

TA: Trusted Authority

TCB: Trusted Computing Base

Acknowledgements

Thanks to my supervisor, Dr. Carlisle Adams, who leads me into this interesting research topic. And also thanks to my family who supports me without any condition.

Chapter 1 Introduction

1.1 The E-Hospital Environment

Nowadays, more and more hospitals are using electronic systems to improve their working efficiency and healthcare services for patients. At the same time, with the paper based patients records being transferred to electronic based, patients' privacy protection is becoming more important. Without proper protection, patients' records can be accessed by unauthorized users or malicious attackers. Sometimes malicious attacks can make the whole system un-accessible. This thesis is about how to design and implement the security features for an e-hospital research project --- MET.

This MET (Mobile Emergency Triage) research is about creating a methodological framework for anytime- and anywhere-decision support (A3Support) for Emergency Department (ED) triage decision-making [36]. The triage, as an Emergency Department (ED) activity, extends beyond the initial assessment and categorization typically completed by a triage nurse to include the initial assessment and management decisions made by the emergency physician (EP). The triage decision-making process involves gathering and evaluating information about a patient and applying medical knowledge to decide on a course of definitive management. This may include further investigations and/or several therapeutic options. The triage disposition may result in discharging the patient with non-serious complaints, or may involve diagnostic tests, specialty consultations, and interim management, which may lead to a definitive diagnosis. Figure 1 is an illustration of the management flow of this e-hospital system.

This e-hospital application is a multi-purpose clinical decision support system and is currently used in paediatric asthma. Data mining techniques are used in this e-hospital system to provide the fast and possible solutions to the ED doctors and nurses. Doctors and nurses can access this e-hospital application system through wired or wireless networks. The accessing devices used by the doctors and nurses to access the system can be PCs, laptops, or mobile phones.

The whole application architecture design follows the management workflow of this e-hospital. There are three components in the application system: user interface, task components, and health information data. The user interface can be in the form of portable computers, mobile phones, or PDAs. The task component, also called a service provider agent, is to implement the end user's request and provide feedback results to the end user. Health information data are generally from hospitals and include the electronic health record, laboratory information system, and admission-discharge-transfer (ADT). All of the health information data are confidential, which means all data in the system have the same security label, namely confidential, and the same security protection mechanism should be applied, if any.

In the function implementation, multi mobile agent architecture has been adopted in this e-hospital application system. This multi mobile agent is inspired by the RETSINA (Reusable Environment for Task-Structured Intelligent Networked Agents (RETSINA)) by Carnegie Mellon University. The structure of the application system reflects the management workflow in the following way: each user group has a corresponding interface agent; each patient management task has a corresponding task agent; each hospital system has a

corresponding information agent. The implementation environment is JADE, which is an agent-orientated language developed in Java. JADE is a middleware for the development of peer-to-peer intelligent-agent applications. It runs seamlessly in the mobile and in the fixed environments, which enables multi-party applications, pro-activity[1], and the machine-to-machine paradigm.

1.2 Problem Description

With rapid growth of networking, in particular the Internet, a large variety of information could be accessed by different users all over the world through a number of software systems. However, the development of corresponding security mechanisms does not parallel the development of the networks and software systems. Many attacks occur and cause big losses. Therefore, security has become a major concern in today's software systems, especially distributed systems. We approach the security problems in this e-hospital system from two levels. First we provide a general overview of the possible attacks existing in software systems and then we describe the specific security problems in a mobile agent system.

In order to find out security problems in a software system, security objectives should be identified. There are several small and large objectivities of a security program, but the main three principles in all information systems are confidentiality, integrity and availability. They are referred to as the CIA triad. All security controls, mechanisms and safeguards are implemented to provide one or more of these principles and all risks, threats and vulnerabilities are measured in their potential capability to compromise one or more of the

[1] Pro-activity: In JADE, agents do not only react in response to external events (i.e. a remote method call) but they also exhibit a goal-directed behavior and, where appropriate, are able to take initiative.

CIA principles. We may also encounter some other security factors such as authentication, authorization, audit, etc., and some of these security factors can be implied by CIA. For example, when considering confidentiality sometimes especially in an application system, we should consider authentication and authorization too. Below is an illustration of the CIA triad.

Based on the security CIA triad, most existing attacks against a software system fall into three categories: attacks against confidentiality, attacks against integrity and attacks against availability. Attacks against availability mainly attempt to make the whole system or part of the system unavailable for the sake of attackers. This kind of attack may shut down the targeted application system or just degrade the targeted system. The notorious attack--DOS (denial of service) belongs to this category. Attacks against confidentiality mainly attempt to reveal the contents of communications, or leak sensitive data and information of a system. Eavesdropping is a primary example of this kind of attack. Attacks against integrity try to modify communication data in a system. Attacks in this category have various forms. Man-In-The-Middle attack, website defacing, and Hijacking attack belong to this category. However, in real world practice, attacks might not just have only one single security objective. Most of the attacks such as virus or unauthorized access might affect system confidentiality, integrity and availability all together.

We talk about the security problems in a general way in the above, now we need to look at the specific security problems in a mobile agent system since this e-hospital application system is a multi mobile agent application system. The mobile agent paradigm is an extension to distributed computing paradigms. A mobile agent is a software program with

mobility which can be sent out from a computer into a network and roam among the computer nodes in the network. It can be executed on those computers to finish its task on behalf of its owner. The direction of a mobile agent can be multi-way, from any computer to another within that network. The migration of a mobile agent can occur multiple times before it comes back to its home computer with the computation results. In addition, not only the application logic of a mobile agent is transferred between computers, but the application state can be transferred from one computer to another. The transferring of a mobile agent state facilitates it in working autonomously to travel between one or more remote computers.

Mobile agent systems have two main components: the agent and the agent platform. An agent has its code and state information to carry out some computation. The agent platform provides the computational environment for agents to operate. The agent platform where an agent originates is called its home platform. The home platform is supposed to be the most trusted environment for the agent. When an agent moves from platform to platform, its code and sometimes its state information move as well.

Although the mobile agent technology has notable advantages over traditional client/server system, it also brings new threats to the system because of agents' mobility. The mobile agent characteristic makes an agent vulnerable to the attacks from malicious agents, platforms or third parties. In addition, since mobile agents have some unique characteristics such as the mobility of a mobile agent, security problems become more complicated in mobile agent systems. The security problems mentioned above cannot be addressed by the protection mechanisms used in traditional client/server system. This broadens the threats in the mobile agent system significantly. Those security problems have become the bottleneck

of the development and maintenance of mobile agent technology, especially in security sensitive applications such as e-hospital.

Four threat categories are identified: threats stemming from an agent attacking an agent platform, an agent platform attacking an agent, an agent attacking another agent on the agent platform, and other entities attacking the agent system. The last category covers the cases of an agent attacking an agent on another agent platform, and of an agent platform attacking another platform, since these attacks are primarily focused on the communications capability of the platform to exploit potential vulnerabilities. The last category also includes more conventional attacks against the underlying operating system of the agent platform.

In this e-hospital application system, all the security problems mentioned above should be considered and possible countermeasures should be given. Since this e-hospital is an electronic medical care system, it also must meet all the security objectives required by related regulations or acts for patients' privacy protection. In this e-hospital application system, there is another special requirement for the access. All the access to the system is determined by hospital's policies. For example one of the policies can be doctors from department 1 can read patients' history records from 9am to 5 pm during the week. When designing the security features to protect this e-hospital application system, all security problems mentioned above and hospital's access policies should be considered.

1.3 Motivation

Over the years, a number of mobile agent systems have been developed and applied. Of all these mobile agent systems, some of them combine security features in their systems and

others do not have security features embedded. The mobile agent system JADE used in this e-hospital is one of the mobile agent systems without many security features in its original version. The security is mainly based on the security features provided by their programming language Java. In a later version of JADE, an add-on security component has been developed which includes authentication, digital signature and encryption. In this security component of JADE, it also has a new JAVA security feature--policy file which can be used to configure access rules. However, the Java policy file cannot make fine-grained access rules.

A lot of research has been done in mobile agent systems security from different aspects. Some of the research work is concerned with the secure execution of a mobile agent in a hostile environment and secure mobile agent migration. And other research work tries to find out a secure framework for mobile agent system in a general way. However, none of them can meet the secure objectives totally in this e-hospital.

We need to provide a secure architecture for this e-hospital system to meet the security objectives. We have noticed cryptography plays an important role in data confidentiality and integrity and it will be used to protect data in this e-hospital. The access control in this e-hospital is based on fine-grained policies. Traditionally access control mechanisms invoke a reference monitor to verify a principal's rights to access a protected object every time the object is referenced. The access control rights of all principals in the system can be viewed as an access control matrix. Reference monitors and access control matrices are originally developed for centralized systems and have their limitations in distributed system.

Since traditional access control mechanism has its limitations in the context of distributed environment, we need to find a possible solution to address the access control

based on policies for this e-hospital. Cryptographic access control seems a promising direction. Cryptographic access control is a new direction in access control research. It eliminates the need for a central reference monitor and it relies totally on cryptography to make sure the data are accessed by authorized person. It is designed to operate in an open network where establishing the identity of the client conveys no information about the likely behaviour of the client and thus is irrelevant to the secure operation of the server. Policy based cryptography seems to provide a possible solution in this e-hospital scenario. It combines data confidentiality, data integrity, access control and network security in one package.

1.4 Thesis Objectives

The main aim of this thesis is to approach the e-hospital system from the perspective of the cryptography. Patients' privacy protection and efficiency are our major concern since some of the end devices in this e-hospital are resource-constrained. Below are the main objectives of this thesis:

- Analyze the application system and find out the possible vulnerabilities and existing countermeasures in the system;
- Analyze the related regulations and laws about patients' privacy protection with the consideration of end users' requirements;
- Define the security scope of the e-hospital based on the analysis of the above two;
- Design the security architecture for the e-hospital to meet the security objective;
- Coding and testing the security features of e-hospital.

1.5 Thesis Contributions

The main contributions of this thesis are as follows:

- We provide a security architecture for a multi mobile agent e-hospital system to meet its security objectives;
- Pairing-based cryptography has been studied for years and a lot of schemes have been designed to overcome the tedious key management in traditional PKI while maintaining the benefits of traditional PKI. We follow the previous work and design the two party key agreement based on identities in this e-hospital system;
- In a centralized environment, access control requires a central reference monitor which may not be practical for distributed systems. Policy-based access control used to be implemented by setting coarse rules through network devices. We try a different solution from the point of view of the cryptography. We combine policy-based cryptography and identity-based cryptography to gain fine-grained access control.
- We extend the policy-based cryptography to propose our extended policy based cryptographic scheme. Our scheme can not only be used in end user access control but also addresses the non-repudiation of end users which is a problem in the original Policy based cryptography.
- Our extended policy based cryptographic scheme has two versions: the basic scheme and the full scheme. The full scheme is chosen ciphertext attack secure under two kinds of adversary. We give security proof to support why it is chosen cipher attack secure.

- We give detailed analysis how our security architecture meets the security objectives of this e-hospital.
- We use pairing-based cryptography to meet most of security objectives in this e-hospital system such as data confidentiality, non-repudiation and network security.
- We implement the security features in java and test the security features and the performance of our proposed cryptographic scheme.

1.6 Thesis Organization

This thesis is organized as follows:

Chapter 1 provides a brief introduction of a multi agent e-hospital, its workflow, components and functional implementation. We give a view of possible attacks against mobile agent systems and related research work on their countermeasures. All the existing research work currently cannot meet the security objectives of this e-hospital, therefore we need to provide a secure architecture for this e-hospital and we also need to address the access control problems in this distributed e-hospital system. Cryptographic access control is a trend in access control and we believe the policy based access control can meet the requirements of this e-hospital very well since all the access is based on fine-grained policies.

Chapter 2 discusses background knowledge and related work regarding mobile agent systems, elliptic curve cryptography, pairing-based cryptography, policy based cryptography, identity based cryptography, access control and database security. All these issues are necessary for the investigation of the proposed framework. This chapter is intended to

establish a firm understanding of security issues in mobile agent systems and to survey the current state-of-the-art technology in existing related research.

Chapter 3 provides an overview of the regulations and laws about patients' privacy protection. End users' requirements have been listed as well. We give details regarding how we approach the security problems in this e-hospital. Based on our analysis we introduce the security architecture for this e-hospital and how to integrate the security architecture into the application system. In this section the details of cryptographic schemes, their security proofs and their algorithms have been listed as well.

Chapter 4 introduces the mathematical background of elliptic curve and pairing-based cryptography. It also includes the description of the Tate pairing and the related algorithms to implement our proposed cryptographic scheme. The test results are given following the algorithms. The tests are done using small prime numbers and large prime numbers. Performance of encryption and decryption under different lengths of policies is also tested in this chapter.

Chapter 5 summarizes the work done in this thesis, discusses some future research directions, and concludes the thesis.

Chapter 2 Related Work

2.1 Mobile Agent System Security

A mobile agent system consists of two main components: mobile agents and their platforms. Gray et al [21] define a mobile agent as follows: A mobile agent is an executing program that can migrate, at times of its own choosing, from machine to machine in a heterogeneous network. On each machine, the agent interacts with stationary service agents and other resources to accomplish its task. Mobile agent platforms are the executing environments for mobile agents on different computers, including the home platforms and guest platforms. Mobile agent technology provides various advantages such as more efficient bandwidth usage, better mobile computing and balancing, etc., which makes it a promising area.

As stated in the introduction, even with outstanding features mobile agents offer a greater opportunity for abuse and misuse, which broadens the scale of threats significantly. A lot of research has been dedicated to addressing the security problems in a mobile agent system. This research differs in its aim, emphasis, base and technique: some propose introducing security mechanisms into the architectures of mobile code systems; some implement real applications with security features; and others are focused on providing a formal framework for a secure mobile agent system. Chess et. al. summarized the assumptions violated by mobile agent systems that underlie most existing computer security implementations such as authentication, reputation and trust [12]. Farmer, Jansen, Rothermel,

et al., analyzed mobile agent system, overviewed the threats and security problems and identified security objects and requirements in [34].

The security problems in a mobile agent system can be divided into four areas: inter-agent security; agent-platform security; inter-platform security; and security between platforms and unauthorized third parties. The research work in mobile agent systems more or less falls into the above four areas.

Over the years, a number of mobile agent systems with security features have been developed and applied. Most of the secure mobile agent systems are trying to address the security issues like authentication, authorization, privacy and confidentiality. For example: Agent Tcl developed by Dartmouth College[22] uses public-key cryptography to authenticate the identity of the mobile agent but the authorization is based on access control lists and at a coarse granularity; Aglet [30] of IBM uses a database to implement a relatively fine grained access control but the authentication is implemented by Message Authentication Code; Ajanata [32] and Ara [37] use domain and place to provide protection for mobile agents; Condordia [43] created by Mistubishi Electronic Information Center America has a strong focus on security and reliability. Cryptography has been used in authentication and mobile agents' migration.

JADE, used in the implementation of functions in the e-hospital system studied in this thesis, is a popular multi mobile agent development platform. The initial software development, that eventually created the JADE platform, was started by Telecom Italia (formerly CSELT) together with the University of Parma in July'98, motivated by the need to validate FIPA specifications [3] through a concrete compliant implementation. Security features were not included in the first version of JADE. In release 3.2 and later versions of

JADE, the JADE board developed a security add-on, JADE-S. JADE-S supports user authentication, agent action authorization and message signature and encryption [27]. Authentication in JADE-S guarantees that the user starting a JADE platform, thereby generating agents, is considered legitimate by having at least a valid identity and passwords in the system. Every agent is related to a user and all the actions of this agent are related to the access rules for its corresponding user. Since signing or encrypting a message slows down the performance of the system, it is the responsibility of the agent to ask the platform to sign or encrypt the message it sends. Though the current version of JADE-S can provide security features, it cannot support the mobility of agents.

SAgent framework in [41] is a secure framework developed for JADE. It is designed to protect the computations of mobile agent applications in a potentially hostile environment. SAgent provides a framework such that applications and security protocol development can be totally independent of each other. In SAgent data format is intermediate and application independent. The application developer is responsible for translating application-specific data to the intermediate format, and the security protocol provider provides routines to convert this intermediate data to a secure format. Data can then be operated on in this secure format by going through methods in the generic public interface, which resolve to the appropriate protocol-specific methods.

In SAgent two secure protocols, namely ACCK and TX, have been implemented. The ACCK was based on Yao's encrypted circuit and two party secure function evaluation protocol [44]. In ACCK, a trusted third party (TTP) is used to perform decryption for remote hosts. The TX protocol proposed by Tate and Xu [40] is less restrictive than the ACCK. TX protocol is a modified ACCK protocol. In the TX protocol, threshold cryptography and secret

sharing were used instead of a trusted third party. For a while, it was thought that the problems of agents against malicious hosts can only be solved through hardware based protection. Tate and Xu's work proves that the hardest problems in mobile agent system can be solved by using software based cryptography as well and their work provides a promising research direction in mobile agent security.

2.2 Cryptography

In this section, we introduce the basic concept of cryptography and related cryptographic scheme on which our scheme is based. We start with the concept of symmetric and asymmetric. We then move to elliptic curve cryptography, pairing based cryptography, identity-based cryptography and policy-based cryptography. We also give an overview of policy based access control and database security.

2.2.1 Introduction

Cryptography is a method of storing and transmitting data in a form that only those it is intended for can read and process. Cryptography provides protection for sensitive information by encoding it into an unreadable format. Cryptography is also a very useful tool for protecting sensitive data as they are stored in media or transmitted over networks. Cryptography has been used to provide the following information security requirements:

- Non-repudiation: Preventing an entity from falsely denying previous commitments or actions.
- Integrity: Ensuring no unauthorized alteration of data
- Authentication: Verifying an entity's identity

- Confidentiality: Protecting the data from all but the intended receiver.

Cryptography may be divided into two categories: symmetric and asymmetric. These terms define whether the cryptosystem uses a single key or a pair of keys for encryption and decryption.

In symmetric cryptography, both parties use the same key for encryption and decryption. The symmetric key is called the secret key between two communicating parties and the security of symmetric cryptography relies totally on how well each communicating party protects this shared secret key. Each pair of users who want to use symmetric cryptography must have their own keys. Symmetric cryptography has the advantage of much faster computing compared with asymmetric and therefore is used to protect large data. But key distribution and management is a problem. Symmetric cryptography can only provide data confidentiality and data integrity; non-repudiation cannot be achieved using symmetric cryptography. Examples of well-known symmetric algorithms include the Data Encryption Standard (DES), Triple-DES, RC4, RC5,RC6 and AES.

In asymmetric cryptography, a pair of keys is used to encrypt and decrypt. Each entity has its own pair of keys and these two different asymmetric keys are mathematically related. If the message is encrypted by one key the other key is required to decrypt. The asymmetric cryptography works more slowly than symmetric cryptography, but it can provide confidentiality, integrity, authentication and non-repudiation. In asymmetric cryptography, key distribution and management are much easier compared with symmetric cryptography. Well-known asymmetric algorithms include RSA, Diffie-Hellman and El Gamal.

The notion of public key cryptography was introduced by W.Diffie and M.E. Hellman in 1976. Their research allowed users to handle key distribution electronically and is known

as the Diffie-Hellman key exchange protocol. Public key cryptography is used widely in industry. The first practical realization of public-key cryptography is the well-known RSA cryptosystem, proposed by Ron Rivest, Adi Shamir and Len Adleman, in which the security is based on the intractability of the integer factorization problem.

2.2.2 Elliptic Curve Cryptography

In this section, we introduce another way of doing public key cryptography—elliptic curve cryptography. Elliptic curves have been studied for years but they haven't been applied in cryptography until recently. Elliptic curve cryptography was first proposed by Neal Koblitz and Victor Miller in 1985. Since then a lot of research on the security and implementation of elliptic curve cryptography has been conducted and several standards have come out. We list several elliptic curve cryptography related standards below:

- IEEE 1363: ECDH, ECDSA,ECMQV and ECIES have been covered in this standard.
- ANSI X9.62 AND X9.63: ECDSA is covered in X9.62, ECDH, ECMQV and ECIES are included in X9.63. Those two standards specify the data format and list recommended curves.
- FIPS 186.2: This is a NIST standard for digital signatures. It specifies DSA and ECDSA and gives a list of recommended curves, which are mandated for use in US government installations.
- SECG: This standard was written by an industrial group led by Certicom. The contents are more or less similar to ANSI and can be accessed from the following website: http://www.secg.org/.

Elliptic curve cryptography provides the same functionality as RSA but the security is based on the hardness of another problem, namely the elliptic curve discrete logarithm problem (ECDLP). Using DLP for public-key cryptosystem and digital signature was first proposed by Taher ElGalmal in 1985. DLP can be described in the abstract setting of a finite cyclic group. Suppose (G,*) is a multiplicative cyclic group and let g be an element of group G. Element g is called the generator of G if the order denoted n of g is also the order of the group G. This means that repeated exponentiation of g will yield all elements of G. Now select a private key x randomly from [1,n-1] and compute public key $y= g^x \pmod p$ where p is the characteristic of the field where G is defined. The problem of determining x given g, n, p and y is the discrete logarithm problem. The operation of $g^x \pmod p$ can be computed efficiently, but it is not practical to calculate x given g, p and y for large p. This is why the DLP is a one-way function and considered a hard problem.

The Elliptic Curve Discrete Logarithm Problem is similar to the aforementioned Discrete Logarithm Problem. The difference is that in the ECDLP, the group is composed of points on an elliptic curve over a finite field. ECDLP is thought to be harder than DLP and therefore elliptic curve cryptosystems have a greater strength-per-key-bit than their discrete logarithm counterparts. The elliptic curve can obtain the same security level with a much shorter key size compared with RSA or DSA. This attribute makes it well suited for systems with constraints on processor speed, heat production, power consumption, bandwidth, and memory. Cell phones, PDAs, wireless devices, laptops, and smartcards are applications that benefit from elliptic curve cryptography.

Elliptic curves used in cryptography are typically defined over two types of finite field: fields of odd characteristic denoted GF(p) and fields of characteristic two denoted $GF(2^k)$.

The elliptic curve group over $GF(2^k)$ consists of the point at infinity ∞, and all points (x,y), where $x,y \in F(2^k)$, that satisfy the equation: $y^2 + xy = x^3 + ax^2 + b$ where $b \neq 0$ and $a,b \in F(2^k)$. The elliptic curve defined over prime field F(p) is defined by the equation: $y^2 = x^3 + ax + b$. In order to form a group on the elliptic curve consisting of integers, $x^3 + ax + b$ can contain no repeated factors. This can be guaranteed by ensuring that $4a^3 + 27b^2 \pmod{p} \neq 0$. All points (x,y) that satisfy the equation $y^2 = x^3 + ax + b$ where $x,y \in GF(p)$, together with an extra point ∞, create a group on the curve.

2.2.3 Pairing Based Elliptic Curve Cryptography

Pairings in elliptic curve cryptography use functions which map a pair of points on an elliptic curve to an element of the multiplicative group defined over a finite field. Let G1 be a cyclic additive group generated by g with order q and let G2 be a cyclic multiplicative group with the same order; a pairing can be expressed by the following function e: $G1 \times G1 \rightarrow G2$. The pairings used in cryptography satisfy the properties below:

- Bilinearity:
  - For all $P, P', Q, Q' \in G1$, we have $e(P+P', Q) = e(P,Q)\, e(P',Q)$ and $e(P, Q+Q') = e(P,Q)e(P,Q')$.
  - For any $P,Q \in G1$ and $a,b \in Z_q^*$, we have $e(aP,bQ)=e(P,Q)^{ab}$
- Non-degeneracy: For all $P \in G1$ where $P \neq 0$, there is some $Q \in G1$ such that $e(P,Q) \neq 1$.
- Computable: there is an efficient algorithm to compute e(P,Q) for all $P,Q \in G1$

The security of pairing-based elliptic curve cryptosystems relies on three hard problems namely the Bilinear Diffie-Hellman (BDH) problem, the Decisional Bilinear Diffie-Hellman

(DBDH) problem and the Gap Bilinear Diffie-Hellman (GBDH) problem. The definitions for these three problems are given below:

- BDH Problem: Given g, g^a,g^b, $g^c \in G1$, where g is a generator of G1 and a,b,c are randomly chosen from Z_q^*, compute $e(g,g)^{abc} \in G2$.
- DBDH Problem: Given g, g^a,g^b, $g^c \in G1$, $r \in G2$ and a,b,c are randomly chosen from Z_q^*, decide whether $r= e(g,g)^{abc}$
- GBDH Problem: For randomly chosen $a,b,c \in Z_q^*$, given g, g^a,g^b, $g^c \in G1$, compute $e(g,g)^{abc} \in G2$ with the help of a DBDH oracle which answers whether a given tuple is a BDH tuple or not.

The most well-known pairings used in cryptography are Tate pairings and Weil Pairings. The Tate paring was introduced to cryptography by Frey and Ruck [15] who considered the Tate pairing over finite fields. The Weil pairing was first introduced to cryptography by Menezes, Okamoto and Vanstone who used it to attack the elliptic curve discrete logarithm problem on certain elliptic curves. Pairing-based cryptography has gained a lot of attention in recent years. A lot of pairing-based schemes, protocols or infrastructures, such as the three party key agreement protocol, identity-based encryption and the short signature scheme, use Tate or Weil pairings in their constructions. We give a brief view of pairing-based cryptosystem below.

Sakai, Ohgishi and Kasahara in [31] proposed pairing-based non-interactive key exchange and signature scheme. The nature of their schemes is identity-based; Joux in [29] put forward a three party key agreement protocol which requires only one round of communication before a shared secret is established. Before Joux's work, key agreement protocols based on Diffie-Hellman needed more than one round of data exchange. In early

2001 Boneh and Franklin[7] proposed identity-based encryption following the work of Shamir in 1984 [39]; Verheul in [42] showed how to use pairings to construct a non-repudiable signature and escrowable public key cryptography using only a public key. In his scheme, the user A chooses a secret $x_a \in Z_q$ and the corresponding public key is $y_a=e(P,P) \wedge x_a$. A CA issues a certificate on the value of y_a (Note his scheme is not identity-based). To encrypt the message M for A, the sender chooses a random number $t \in Z_q$ and generates the ciphertext:

$$C=<U,V>=<[t]P, M \oplus H_2((y_a)^t)>$$

where H_2 is a hash function H_2: $G2 \rightarrow \{0,1\}^n$ and n is the bit length of message space. To decrypt, the end user A computes

$$M'=V \oplus H_2(e(P,U) \wedge x_a).$$

The escrowable service is supported as follows. Ahead of time, the user A sends the escrow agent the value $[x_a]P$. The escrow agent can compute $e(P,U) \wedge x_a$ using his knowledge of $e(y_a, U)=e([x_a]P,U)=e(P,U) \wedge x_a$. The user A does not need to give his secret to the escrow agent, and thus A's signature remains non-repudiable.

A lot of pairing-based signature schemes have been proposed over the years. One of them is the short signature by Boneh, Lynn and Shacham [6]. In one version of their signature scheme, a 170 bit signature offers security comparable to that of a 320-bit DSA signature. Pairing-based cryptography also found application in network security. Kempt et al.[18] used IP address as an identity and added identity-based signatures to routers to address security in IPV6. Khalili et al. [3] combined identity-based cryptography with threshold cryptography to build a key distribution mechanism for ad-hoc networks.

Appenzeller and Lynn[2] used the non-interactive key distribution scheme (NIKDS) of Sakai et al. to produce identity-based keys for securing IP packets between hosts. Smetters and Durfee[13] proposed a system in which each DNS runs its own IBE scheme and distributes private keys to its hosts in the domain.

2.2.4 Identity Based Cryptography

Identity-based cryptography was first proposed by Shamir in 1984. The basic idea is to simplify the key management of some Internet applications using public key cryptography. It works in the following way: Suppose Bob wants to send some confidential information to Alice. First, Bob will encrypt the confidential information by a symmetric key and then he will encrypt this symmetric key using Alice's identity, such as her email address. The email address here is Alice's public key and only Alice has the corresponding private key. After receiving the encrypted message, Alice uses her private key to decrypt and get the symmetric key. She can then use this symmetric key to decrypt the encrypted message. The channel through which Alice gets her private key from a private key generator (PKG) is supposed to be a secure one. Identity-based cryptography provides a secure way to transmit a session key or a symmetric key over an insecure communication channel such as the Internet.

Identity-based cryptography is a form of public key cryptography, but it is different from traditional PKI in which both parties should have their private keys before communication occurs, and key management is much easier than that in traditional PKI. But in identity-based cryptography, since the public key is the users' identity, key revocation can be a problem. And in identity-based cryptography, the PKG knows the private keys of users

and because of this non-repudiation cannot be achieved and there is an inherent key escrow mechanism in identity-based cryptography. In traditional PKI, the CA doesn't need to know users' private keys and non-repudiation is possible. Identity-based cryptography has generated a lot of research interest but it doesn't mean identity-based cryptography has more to offer than traditional PKI. Both of them have their own benefits and drawbacks and which to use is dependent on the application. For some applications that require non-repudiation, traditional PKI is more suitable while identity-based cryptography may be more useful in the scenario of an enterprise that needs recovery of encrypted documents or emails.

Some research has been done in combing traditional PKI with identity-based cryptography to get the benefits of both. Chen et al. [10] proposed a mixed architecture with identity-based TAs administering users at the lowest level of a hierarchy supported by a traditional PKI. Gentry introduced the idea of Certificate-Based Encryption (CBE) in [19]. The aim of CBE is to simplify the revocation in traditional PKI and use pairings to construct the encryption scheme. In the CBE scheme, a user's private key is composed of two components: the first part is issued as a certificate by a traditional CA and is time-dependent; the second part is a private key derived from the user's identity. The only problem in this CBE scheme is that the CA needs to issue new certificates to users in each time period. Al-Riyami and Paterson [1] proposed certificateless public key cryptography (CL-PKC). The users' private key generation consists of two stages. In the first stage, the userA gets the identity-based private key from a key generation center (KGC) in a secure channel. In the following stage, the user generates his/her private key using the key from the first stage and a secret chosen by A. The public key of A is a key pair designed from the above two stages and the validity of the public key can be checked by comparing the public key pair. In CL-PKC

the validity of the public key can be checked without the participation of the CA as in traditional PKI. Details of CL-PKC can be found in chapter 3.

Identity-based encryption has been developed and used in industry. Voltage Security is a company who is actively engaged in developing identity-based encryption systems. Voltage Security has identity-based email and file encryption systems. In June 2008, Trend Micro, a well-known antivirus vendor, launched its new identity-based email encryption solution. This email encryption is integrated into its antivirus clients. The email encryption solution of Trend Micro not only supports identity-based encryption but also can support policy based encryption.

2.2.5 Policy Based Cryptography

Policy-based cryptography is similar to identity-based cryptography. The difference is that in policy-based cryptography, a specific policy or set of rules is used as the recipient's public key. Only those who are compliant with the policy can decrypt the message. Bagga and Molva propose a policy-based encryption system [4]. In their system, policies are formalized as monotonic logical expressions involving disjunctions (denoted $\vee$) and conjunctions(denoted $\wedge$) of conditions. Each condition is fulfilled by a specific credential issued by a certain trusted authority TA. An entity fulfills a policy if and only if it has access to a set of credentials associated with the logical combination of conditions defined by the policy.

Policy based cryptography can be used to provide data confidentiality and access control simultaneously. Policy based cryptography can be used in the scenario where the access

control rules are complicated and hard to be implemented by rules of network devices or application systems. For example if the access control policy of a certain type of data "res" of company A is: "res" can be read if the user is from company B , he should be a member of organization C, the access time is between 9am and 5pm and he must be in place D. This policy is hard to be fulfilled by setting rules in the network devices or database and it may involve cross certificates of multiple companies in a traditional PKI system. Policy based cryptography provides a possible solution to the case above.

Bagga and Molva also outlined several scenarios where policy based cryptography can be used. Imagine there is a pdf file containing confidential information and Alice wants to read it. The pdf file owner Bob wants to know whether Alice complies with the read policy of this file or not. In other cryptographic schemes, Alice should show her credentials to prove she complies with the policy but in doing this, some Alice's private data could be disclosed as well. In policy based cryptography, Bob just needs to encrypt the pdf file using the policy as the key and Alice uses her corresponding credentials to decrypt. Here the real encryption of the pdf file is done by a symmetric key and the policy is used to encrypt the symmetric key. The scenarios here can be used to solve the cyclic policy interdependency problem described in [8]. Another application of policy-based cryptography is the sticky privacy policy paradigm in which policy is specified, agreed to by data owners, and governs the data lifetime. Here policy is used to reflect data owners' privacy preferences and encrypt the data. Only those who fulfill the privacy requirements can decrypt and get the data.

In a client-server environment, there is a central reference monitor mechanism in access control. But with the transition of the client-server paradigm to distributed systems, the

access control enforced by the central reference monitor mechanism may not be attained. Using cryptography in access control provides a promising direction. In [25], Wilkinson et al. show how to use cryptography to achieve trustworthy access control in an untrustworthy web server. The access control functionality is fulfilled by an encryption/decryption proxy. In their proposal, the access control policies are described through conjunctions and disjunctions of groups each containing a number of users. They describe how to encrypt and generate decryption keys corresponding to the access control policies. They use traditional PKI in their proposal and therefore it involves public key certificates. In [24], Harrington, Anthony, et al., propose a cryptographic access control for a distributed file system. They show how to use encryption/decryption keys to implement read and write access in a distributed file system.

V. Goyal, O. Pandey et al. in [20] proposed attribute-based cryptography to get fine-grained access control. Their work was based on the attribute-based encryption work of Sahai and Waters[38]. Sahai and Waters first introduced the concept of attribute-based encryption (ABE). In ABE, users' keys and ciphertexts are labeled with descriptive attributes. If there is a match between the descriptive attributes of users' keys and ciphertexts, users can decrypt. Goyal, Pandey, et al., extended ABE and proposed a Key-Policy Attribute-Based Encryption (KP-ABE). In KP-ABE, ciphertexts are labeled with attributes and users' private keys correspond to an access structure with leaves associated with attributes. The users' private key structure specifies which ciphertext users can decrypt. The interior nodes of private keys' access structure consists of logical "And" and "Or" gates. One of the applications of KP-ABE is targeted broadcast. Each broadcasting program is labeled with attributes and users'

private keys are associated with an access policy. Users subscribe to different broadcasting packages and decrypt using their private key constructed by access policy.

2. 5 Summary

In this chapter, Mobile Agent Security problems and related research work have been introduced. We look at the mobile agent development system--JADE used in this e-hospital and related security research in JADE. We give a brief overview of elliptic curve cryptography and pairing-based elliptic curve cryptography. We investigate the related research in elliptic curve based cryptography such as identity-based cryptography, policy-based cryptography, their applications and cryptographic access control.

Chapter 3 E-Hospital Security Architecture

This chapter presents the design of the security architecture for this e-hospital. We use ISO17799, one of the most accepted international standards in information security management, as guidance in the design. Security is approached in the following way. First, security objectives are reached by the consideration of the hospital's patients' privacy protection policy, the related patients' privacy protection laws, regulations and risk assessment of the current e-hospital. The risk assessment is to identify the possible vulnerabilities and threats to the e-hospital and is based on the analysis of the application framework of this e-hospital. Secondly, the whole security architecture has been given, followed by a discussion of the specific cryptographic components. Finally, a security analysis has been done to show how our design can meet security objectives of this e-hospital and to demonstrate the security of our proposed cryptographic scheme.

3.1 Approach

A successful security architecture includes a heterogeneous combination of policies and leading practices, technology, and a sound education and awareness program. A security architecture provides a scalable framework for integrating people, processes, and technology related controls that addresses both current and planned business objectives. A good security architecture is not only a reflection of business requirements but also helps to manage the complexity of the business.

Over the years, many security standards in information security management systems have been published, of which ISO17799 is one of the most accepted standards. ISO17799 is

a code of practice for information security management that has been published by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC). Since its publication, ISO17799 has gained universal recognition as one of the most widely recognized descriptions for an information security framework. Requirements described in ISO17799 are already expressed in cooperate security policies of many leading organizations. We follow the ISO17799 as our guide in the security architecture design. We also refer to ISO27799 which provides guidance to apply ISO17799 when securing a health information system or protecting personal health information.

ISO17799 is based on assuring the confidentiality, integrity and availability of information assets, and integrity and availability of the supporting systems. Controls have been used to gain assurance. There are 11 key control areas to make a successful security program in ISO17799. Those 11 control areas have been broken into 39 categories and for each category there is a security objective and corresponding controls to gain the security objective.

Below is a brief summary of those 11 control areas along with their corresponding security objectives.

- Information security policy provides requirements for the development of security policy documents. Its purpose is to provide management direction and support for information security policy.
- Organizational Security provides requirements for the establishment and organizational structure of a security program. Its objectives are to establish a management framework to implement information security within the organization,

maintain the security of facilities and information assets accessed by third parties and ensure information security when it has been outsourced.

- Asset Classification and Control provides requirements for asset classification and protection for those assets. Its aim is to classify assets including information, software, hardware and utilities and ensure those assets receive the appropriate level of protection.
- Personnel Security provides requirements for personnel management, end user awareness training and incident response procedures. It aims to reduce risks caused by human beings, minimize the damage from security incidents and ensure end users' awareness of information security.
- The aim of Physical and Environmental Security is to prevent unauthorized access, damage and interference with information, prevent loss and damage to assets or business activities, and prevent the compromise or theft of information or facilities.
- Communication and Operations Management provides requirements for the management of operations of systems and electronic communications. It is concerned with change management, incident management, separation of duties, facilities management, malicious software defense, system backup and recovery, audit logs, network management, media management and information exchange between organizations.
- Access Control deals with directories, operation systems, application systems and network infrastructure access management. Its aim is to control access to information, prevent unauthorized access, protect network services, prevent

unauthorized computer access, detect unauthorized activities and ensure mobile computing security.

- System Development and Maintenance provides requirements for system development and information technology infrastructure. It includes how to document security requirements for systems or application security controls, message and data validation, cryptographic controls, digital signature, key management, system file security and integrity, change control processes and malicious software.
- Business Continuity Management provides requirements for business continuity planning and disaster recovery planning. Its objective is to protect critical business functions from the effects of failure.
- Compliance deals with legislative and regulatory compliance, intellectual property rights, organizational records, personal privacy, misuse, regulation of cryptographic technologies, evidence handling, security policy review, technical compliance and system audits.
- Incident handling is closely related to disaster recovery planning and should be part of the company's disaster recovery plan. The primary goal of incident handling is to contain and repair any damage caused by an event and to prevent any future damage.

3.2 System Security Analysis

In this section, we give an overview of the current security situation of this e-hospital. We focus on the e-hospital application security; however, we also look at other areas defined in

ISO17799. In particular, we discuss patients' privacy protection regulations and laws, the overall security policy of the hospital, and end users' specific requirements and expectations for this e-hospital system. Based on this analysis, we get a picture of the current security situation of this e-hospital, the possible threats it is facing and what should be done to decrease the possible risks without affecting the availability of this e-hospital application system. At the end of this section we reach the security objectives for this e-hospital.

3.2.1 Hospital Security Policy

The hospital has its own security policy, security of patient personal health information, to protect patients' privacy. In the security policy file, it defines patients' health information, working files, and who bears responsibility for the protection of patients' health information. The security policy also defines the types of protection that should be adopted to protect electronic patients' health information, secure transfer of patients' health information from one area to another, and handle confidential paper waste.

Some of the requirements within the hospital's security policy which can help us to construct security objectives are listed below.

- All health care information of each patient is compiled into one health record with a unique file number.
- Health Records maintain an electronic tracking of the locations of all patient records.
- Health records are controlled centrally by the hospital. Once a patient's record has been loaned, it is the loaner's responsibility to safeguard the record. All patients' records must be returned to central health records daily.

- The transfer of patients' records from one area to another must be protected from unauthorized access and viewing.
- Health records should be held in a designated, secured area and inaccessible to those unauthorized persons but easily accessible to authorized health records personnel.
- Personal health information is any information that identifies the individual or can be manipulated to identify a person such as health card number, the donation of any body part, payments or eligibility for health care and physical and mental health information.
- Working files are not part of the patients' records; they are used and maintained by medical staff to aid their daily work, including items such as reminder notes, contact phone numbers, and physical copies of chart reports.

In addition to the policy mentioned above, there are two important conceptions in this e-hospital practice. The first one is the circle of care which defines members of the health care team who are involved in providing care of treatment to a particular patient. Members of a circle of care can collect and use the patient's personal health information for that care, unless the patient has expressly withheld or withdrawn consent. The second conception is a "Lock Box" which clarifies that a patient has the right to have his/her health information withheld from the members of a circle of care.

3.2.2 Legal Compliance

With the increase of medical breaches in North America, legislation and laws have been created to make sure medical care providers meet the requirements to protect the patients' privacy. Of all that legislation, the federal Personal Information Protection and Electronic Documents Act (PIPEDA) and Personal Health Information Protection Act (PHIPA) are

Canadian, and are therefore the most relevant to Canada. The hospital uses these two acts in their legal compliance. PHIPA is a provincial privacy legislation that is similar to the federal privacy legislation PIPEDA. PHIPA is more specific to the health care sector while PIPEDA applies to the whole private sector including pharmacies, laboratories and heath care providers.

Personal Information Protection and Electronic Documents Act (PIPEDA) aims to protect personal information in Canadian commercial activities. It applies to paper-based business and online electronic business activities. The act guarantees Canadians' right to know why their personal information is used, collected or disclosed by organizations. The act defines personal information in a broad way. Any information which can be used to identify a natural human being, as opposed to a legal entity, is considered personal information. The act contains ten guiding principles and related explanations and rules.

Personal Health Information Protection Act (PHIPA) is privacy protection legislation built on the Personal Information Protection and Electronic Document Act (PIPEDA) guidelines for the protection of personal electronic information. PHIPA is informed by 10 principals to protect personal information: namely accountability; identifying purpose; consent; limiting collection; limiting use, disclosure and retention; accuracy; safeguards; openness; access; and challenging compliance [30]. PHIPA defines personal health information (PHI) as identifying information about an individual in oral or recorded form that

- relates to his or her physical or mental health;
- relates to providing health care, including identifying a provider of health care;

- is a plan of service within the meaning of the Long-Term Care Act,
- relates to the donation of a body part or bodily substance,
- relates to payments or eligibility for health care in respect of the individual,
- is a health number,
- identifies a substitute decision-maker of that individual,
- is in a record held by a HIC[2] where the record contains any of the above information

The health level seven (HL7) standard is another guideline which the e-hospital should comply with HL7 specifies the security framework for medical data exchange and addresses the following levels or layers of communication link, network, transport, session, and application For each of these layers of communication, HL7 addresses the following security services authentication, authorization and access control, system and data integrity, confidentiality, accountability, availability, and non-repudiation

3.2.3 Risk Assessment

In this section, the e-hospital application, its physical environment, its platform, its network protocol, its data exchange, its access control and its database system are discussed The vulnerabilities in the current system and the possible risks it might have are identified The risk assessment in this section can help us reach our security objectives of this e-hospital in the following section

[2] HIC Health Information Custodian includes a health care practitioner, a service provider within the meaning of Long Term Care Act, a minister of Health and Long-Term Care, a medical officer of health, and a person who operates a health facility, pharmacy, laboratory etc

This e-hospital adopts a multi agent architecture in its functional implementation. As mentioned in Chapter 1, there are four types of attack in agent based systems: an agent attacking a platform, a platform attacking an agent, an agent attacking another agent, and a third party attacking an agent based system. For this e-hospital, we need to look at its development framework to find out more possible attacks and risks.

JADE is used as the development framework for this e-hospital. JADE provides basic middleware-layer functionality which is independent of the specific application and which is fully compliant with the Foundation for Intelligent Physical Agent (FIPA) specifications. FIPA was established in 1996 as an international non-profit association to develop a collection of standards relating to software agent technology. The most important conceptions of FIPA are agent communication, agent management and agent architecture. JADE fully implements those FIPA conceptions.

Agents in JADE live in containers which are the platforms for agents. Containers provide the JADE run-time and all the services needed for hosting and executing agents. All containers and agents have a unique identifier. The main container is the first container to be launched in JADE and all the other containers must join to a main container by registering with it. The main container is the bootstrap point and it manages the container table (CT) and the global agent description table (GADT); it also hosts the agent management system (AMS) and directory facilitator (DF). The CT is the registry of the object references and transport addresses of all container nodes composing the platform. The GADT is the registry of all agents present in the platform including their status and location.

The AMS and the DF are two special agents providing agent management and white page service, and the default yellow page service of the platform, respectively. The AMS

supervises the entire JADE platform and is the contact point for all other containers in the platform. Every other container must register with AMS to obtain a valid agent identity. The DF implements the yellow pages service and is used by agents wishing to register their services or search for other available services.

The local agent description table (LADT) in other containers synchronizes with the GADT in the main container. When an agent wants to communicate with another agent, it searches the LADT first to locate the recipient and if it fails it will search GADT for synchronization. This kind of mechanism prevents the main container from becoming a bottleneck. There is also a main replication service to ensure the availability of the JADE platform even if the main container fails.

Agents communicate with each other in JADE by asynchronous message passing. Each agent has a 'mailbox' to receive messages sent by other agents. Each message contains a set of one or more message parameters. Most messages contain sender, receiver and content parameters. The encoding for messages can be string, xml and bit-efficient.

JADE uses its own Internal Message Transport Protocol (IMTP) to exchange messages between agents in different containers. For the communication between JADE and non-JADE platforms, the Message Transportation Protocol (MTP) is used. Usually for inter-platform communication, JADE starts an HTTP-based MTP in the main container. MTP is only started in the main container.

There are three types of agents, namely interface agents, task agents and information agents, in this e-hospital. Interface agents are used to interact with users, receive user inputs and display results. Different users, based on their roles such as doctors or nurses, log on to

different interface agents to access the system. Task agents help users perform tasks, formulate problem-solving plans and execute these plans. Information agents provide access to a heterogeneous information source. The access is controlled by two agents: the authorization task agent and the lock box task agent.

The authorization task agent is used to decide which kind of interface agents can access which kind of task agents. End users access the system by providing their username and password and by swiping their work badge card. The lock box agent is used to decide whether the task agent can perform the special task for the specific user. End users can access the e-hospital through wired and wireless networks in the hospital.

Based on the above descriptions, there are several risks in this e-hospital system which are listed below.

- There is no authentication mechanism between the main container and normal containers. Malicious normal containers can attack the main container and other containers in the platform;
- The main container can attack other normal containers if the main container is not managed properly. For example if the main container goes down because of an attack, the malicious container can pretend to be the main container and attack other normal containers;
- All the messages are transmitted in clear text inside the e-hospital and an attacker can sniff the network to get the message;

- The wireless network protocol used in this e-hospital is Wired Equivalent Privacy (WEP). WEP is not a secure protocol. It is vulnerable to social engineering attacks because of the key reuse; and it is also vulnerable to jamming and flooding attacks. There is only one-way authentication and integrity is based on a non-cryptographic checksum in WEP. The RC4[3] algorithm is used in WEP to provide confidentiality. The problem is that RC4 is a stream cipher algorithm which is not appropriate for a wireless medium where packet loss is widespread. To solve this problem the same key is used in WEP which makes it vulnerable to guessing attacks. All these make WEP not appropriate for a confidential application like this e-hospital.

- Confidential information is placed in agents' mailboxes in clear text and attackers may get access to the information.

- Patients' private information is stored in the database in clear text. If by any chance the attacker knows the password to access part of or the whole database, he can get the patients' private data.

3.2.4 E-Hospital Security Objectives

We reach the security objectives of this e-hospital on the basis of the previous two sections. Some of the security objectives have been achieved by the current system such as end user authentication, circle of care, and the lock box, etc. Personnel security such as background checking, end users' awareness training and how to deal with the third party have been

[3] RC4: RC4 is a symmetric stream ciper designed by Rivest for RSA Data Security. The encryption and decryption is performed by combing the keystream with the plaintext by using bit-wise exclusive-or.

covered in the security policy of the hospital, therefore we follow the hospital policy and do not list the personnel security as one of the security objectives of this e-hospital.

Details of the security objectives of this e-hospital are described below:

- Sensitive information should be protected with a higher level of security. Sensitive information includes personal health information, pseudonymous data derived from personal health information via some methodology for pseudonymous identification, statistical and research data, audit trail data and system security data including access control data or system security configuration data;
- Patients' records should be returned to the hospital at the end of the day. Since Interface agents are dealing with end users, patients' information left in the interface agents should be removed at the end of the day;
- According to the hospital security policy, no sensitive data is allowed to be used outside of the hospital. One possible way to take this data outside the hospital is for staff to take out end devices such as laptops or PDAs where interface agents reside. Therefore any sensitive data in interface agents should be removed when the device is not inside the hospital;
- The confidentiality and integrity of the sensitive information should be enforced to make sure unauthorized parties cannot view and modify protected information. This protection mechanism can also prevent some mistakes by authorized staff. For example if an authorized staff member accidentally sends the sensitive data to a wrong recipient, the confidentiality and integrity protection mechanisms can make sure that the unauthorized recipient cannot access the sensitive information.

- In order to confirm that end users access the confidential information, non-repudiation should be provided in this e-hospital. Non-repudiation of this e-hospital needs only to make sure that the sender (the backbone system) can confirm that the intended recipient (the end user) has received the information;
- End users must authenticate to the system before they log on to the e-health system. This should be a strong authentication such as two-factor authentication;
- In order to prevent malicious agents from attacking other agents, agents should authenticate each other before data exchange;
- A normal container and the main container should authenticate each other before the normal container registers to the main container. An illegal normal container is not allowed to register to the main container and an illegal main container is not allowed to provide service to other normal containers;
- Access to sensitive data should be limited to a need-to-know basis. Access should also be based on the fact that individually identifiable health information is collected, accessed, used or disclosed only with a patient's consent;
- Sensitive data in the database should be protected against unauthorized viewing, copying, use, or modification;
- The physical environment of this e-hospital should be protected from attacks. The protection of physical environment can prevent damage, theft and unauthorized access to the facilities such as servers, network devices and end user devices;

- The communication between the e-hospital and other hospitals should be protected against malicious attacks such as network sniffers. According to the HL7 standard, data exchange among hospitals should be encrypted by strong algorithms.

- The system should keep audit trails to identify security relevant events such as possible attacks or unauthorized access. Audit trails also keep track of the viewing, coping and exchange of sensitive information. Audit trails are also part of the sensitive information and should be protected;

- The communication channel between agents should be protected against malicious attacks since the communication protocol between agents is IMTP and messages are transmitted in clear text. Attackers can get sensitive information just by sniffing the network.

- Different agents in this e-hospital are responsible for different tasks and dealing with different data with different security levels. For example, information agents access the data from the database directly and therefore the protection mechanism should not be the same as those for interface agents which are dealing with end users. Therefore different agents, namely interface agents, task agents and information agents, should be in different networks and those different networks should be separated by protection mechanisms;

- The underlying operation system provides the system service for this e-hospital application system and has a higher security level than the application system. If the operation system has been attacked successfully, the security of the application system

will be jeopardized. Therefore, the operation system of this e-hospital should be protected from malicious attacks such as viruses, phishing or unauthorized access;

- According to PHIPA and end users' requirements, medical data should be updated and easily accessed by authorized personnel. This requirement leads to the continuity of operation of this e-hospital. Authorized users are supposed to be able to use the e-hospital system even in the case that attacks happen.

3.3 Security Architecture

In this section, we describe how we design the security architecture of this e-hospital to meet the security objectives and to decrease the possible risks. This section is organized in the following way: in section 3.2.1, we introduce our layer domain strategy to design the architecture; in section 3.2.2, we give descriptions of what kinds of mechanisms are used to protect each domain; in section 3.3.3, we focus on our proposal in cryptography which is based on certificateless cryptography and policy based cryptography.

3.3.1 Layer Domain

When we approach the security architecture for a system, we notice there are different protection mechanisms working at different levels. For example, a security guard can scare away evildoers, but what if an intruder attacks the system from a network? Here, the intrusion detection system can provide some level of security. In order to strengthen the security features of a system, we cannot rely on only one protection method; we should use different layers of security strategy to protect the whole system.

We have noticed in this e-hospital different components have different security levels. For example, the information agents are dealing with the database directly, and therefore the security level of information agents is different from the security level of interface agents which handle requests from end users. Another example is the operation system and the software running on it. The operation system should have a higher security level compared with that of the software and should be trusted more. Following the idea of different security levels with different trust levels, we separate components with different trust levels in this e-hospital into different domains. The layered protection has been adopted inside and among domains.

Our domain based strategy borrows the idea of trusted computing base (TBC) from the Orange book.[4] The TCB addresses the level of trust a system provides. The TBC can be a combination of hardware, software and firmware. The TCB looks at all protection mechanisms within a system to enforce the security policy and provide an environment that will behave in the manner expected of it. The communication between TCB and non-trusted components is controlled by the security perimeter. The security perimeter separates trusted components from untrusted components by providing an interface for those entities to communicate securely.

In the case of this e-hospital, we define three domains, namely the trusted domain, the security perimeter domain and the untrusted domain. The trusted domain has the highest trust and security level. If the trusted domain has been successfully intruded upon, the whole system security will be jeopardized. The untrusted domain has the least trust level and is

[4] Orange Book is the Trusted Computer System Evaluation Criteria (TCSEC) published by the U.S. Department of Defense. The criteria are published in a book with an orange cover, which is called the Orange Book.

separated from the trusted domain by the security perimeter domain. The security perimeter domain provides the communication interface for the trusted domain and the untrusted domain and has a trust level in between those two. Each domain contains JADE components used in this e-hospital, the underlying supporting hardware, operation system and software.

Following the definition above, in this e-hospital the information agents, database and the main container can be thought of as the trusted components and are defined to be in the trusted domain. The task agents, authorization agents and lockbox task agents fall into the security perimeter domain which separates trusted domain and untrursted domain and which provides the secure communication interface for those two domains. The user interface agents are dealing with end users and more exposed to attacks than other domains. They should be put into the untrusted domain with the lowest trust level. The untrusted domain cannot access the trusted domain directly and vice versa. All the communication should be monitored by the security perimeter domain in a secure way. The access control among domains is illustrated in figure 3-1.

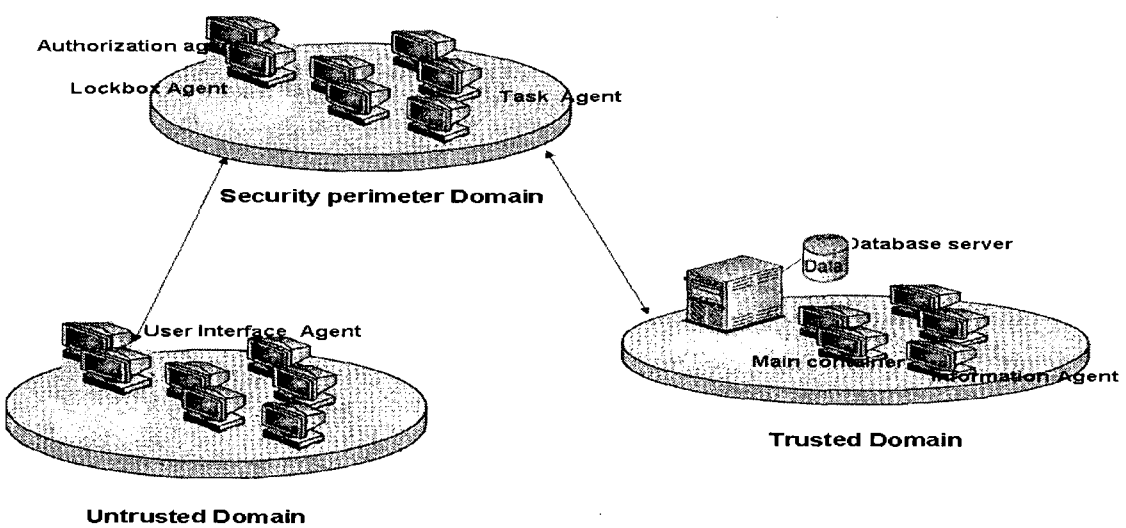

Figure3- 1 the Access Control among Domains

Now we have a clear picture of how we design protection mechanisms for the system. For each domain, we need to provide different protection mechanisms for different components working in different security levels. We also need to protect the communication among domains. For the protection of this e-hospital application system in each domain, we propose to use cryptography, which is based on identity based cryptography, to provide authentication, data confidentiality, data integrity, non-repudiation, secure communication channel and access control in one package. For the supporting systems like operation system, network and physical environment, we need to use other protections. The protection mechanisms used in each domain and among domains are described below.

- Protection for the untrused domain
    - End users access the e-hospital system through the untrusted domain. To prevent unauthorized users from accessing the system, we use stronger authentication, which is a two-factor authentication [5], to authenticate end users. Currently username/password and the radio frequency identification (RFID) card are used in the e-hospital as the authentication mechanism. The RFID card is used by hospital staff to access the hospital facilities.
    - User interface agents live in normal containers. To prevent malicious containers or malicious agents from attacking the main container or other agents, normal containers and interface agents must authenticate each other before interface agents can run in the normal containers. As we mentioned before, the name for

[5] Two factor Authentication: There are three general types of authentication something a person knows such as password or PIN, something a person has such as a key, access card or badge, and something a person is which is biometrics Authentication using two out of these three methods is called two factor authentication and is thought be a stronger authentication

each container or each agent is unique and is used by the system to find the container or the agent. We use credentials, which are generated from the name of normal containers or interface agents, as authentication mechanisms. The credentials we use are private keys of identity based cryptography. As introduced in chapeter2, identity based cryptography has an inherent key escrowing mechanism. The private keys can be distributed to end users before or after the communication. In our case, credentials are preinstalled to legal end devices.

- All the information exchanged in JADE is in the form of messages and is transmitted between and among containers and agents in clear text. The underlying network transportation protocol WEP [6] does not provide strong protection against network attackers. We need to protect the information from attacks such as network sniffers[7] by encryption. The encryption key is generated from an identity based two party key agreement protocol which is described in the following section. The credentials used in a two party key agreement protocol are based on the identity of the container or the agent and are the same as we used in the authentication.

- The hospital staff accesses the patients' information according to the policies of the hospital. The access policy is composed from users' group information, the access time information, the access location information and additional constraints such as whether patients have withheld their health information from

[6] WEP Wired Equivalent Privacy is a deprecated algorithm to secure 802 11 wireless networks

[7] Network Sniffers A network sniffer is a tool that monitors traffic as it passes by. Sniffers are also referred to as network analyzers or protocol analyzers When used as a diagnostic tool, it can help a network administrator to trace the root of network problems When it is used as a tool for an attackers, confidential information such as usernames and passwords can be captured as they travel over the network

some members of the group. The access policies are not stable and change with time and the turnover of the staff. Therefore the access policies cannot be easily or practically enforced by configuring access rules on network devices or in the database. We propose to use cryptography to implement end user access control. After the end user authenticates to the system, the credentials corresponding to the policy for the end user are distributed to the end user securely. Only those who have the credentials corresponding to the specific policies can be regarded as legal users and can access the information protected by the policy. The cryptography is based on the policy based cryptography and we extend the policy based cryptography to address the non-repudiation of the end user by introducing a secret only known to him. The end user decrypts by combining his secret and those credentials corresponding to his access policy. The details can be found in the following section.

- In order to ensure non repudiation of the message, we propose a cryptographic scheme based on certificateless cryptography and policy based cryptography. The end user keeps a secret (private key) only known to himself and combines this secret with the partial private keys from trusted authority to construct his final private key. In this way we can reduce the risk of repudiation.

- In order to prevent the attacker from taking advantage of the vulnerabilities of the operation system to attack the e-hospital application system, we need to protect the underlying operation system from malicious codes and unauthorized access. We need to use third party solutions such as antivirus software or personal

firewalls to protect the operation system. We also need to follow the guide from the operation system vendor to configure the operation system securely, stop the unnecessary system services and apply the secure update patches in time.

- Protection for the security perimeter domain
  - The security perimeter domain has a higher security level than the untrusted domain. It monitors all the communication between the trusted domain and the untrusted domain. It protects the trusted domain from the untrusted domain and acts like a buffer zone between trusted and untrusted domains. We need to put these three domains in different networks and enforce the access rules among domains. We use two firewalls to separate the three networks. One firewall is between the untrusted domain and the security perimeter domain and the other one is between the security perimeter domain and the trusted domain. We configure the access rules on the firewalls to make sure the trusted domain only accepts requests from the security perimeter domain and the untrusted domain cannot access the trusted domain directly. The firewall filters out all the packets coming in or out of the network it is protecting and discards, repackages or redirects the packets according to the access security policies.
  - A firewall is good for controlling access, but it cannot protect the network from attacks such as denial of service, port scans or inside attacks. To detect those attacks, intrusion detection systems (IDS) should be used. IDS can be configured to monitor the network or a segment of the network. IDS can be used to warn system administrators of the possible attacks or automatically set the firewall to

block some attacks. Also IDS is a very useful tool to parse audit logs and thus to discover possible attacks.

- As mentioned in the untrusted domain, in order to prevent attacks from unauthorized agents or containers, mutual authentication should be adopted. We use credentials to authenticate legal agents or containers before they can register to the main container of this e-hospital system. Credentials are private keys of identity based cryptography and are generated from the identity of agents or containers. The identity is the name of the agent or the container.

- Task agents, authorization agents and lock agents are in the security perimeter domain. Messages exchanged among agents, containers inside this domain or across domains are in clear text. Therefore we need to protect the messages by encryption. Like the encryption keys used in protecting messages in the untrusted domain, the encryption keys used in this domain are generated from an identity based two party key agreement protocol.

- The security perimeter domain is in the backbone network and only authorized staff such as the network administrator can physically access the equipment in this domain. To avoid possible attacks such as unauthorized access or theft, all the equipment should be locked in the server room of the hospital and be monitored by the surveillance system of the hospital such as closed-circuit TVs (CCTVs).

- To protect the operation system from virus attacks or attacks exploiting operation system's vulnerabilities, we need to use third party antivirus software to protect

the operation system, as well as configure the operation system securely and apply security patches in time to avoid possible attacks.

- In order to trace possible bad deeds, detect intrusion and reduce the risk of unauthorized access in the security perimeter domain, we need to provide accountability in this domain. Accountability is tracked through the recording of system and application activities. We use the system log of the operation system to trace system activities and we record the agents, containers and end user activities in the application audit log. The audit log can only be accessed by authorized programs or staff, and is designed to be read/add events only.

- Protection for the Trusted Domain

  - The trusted domain has the highest security and trust level and only accepts legal requests from the security perimeter domain. As mentioned in the protection for the security perimeter domain, there is a firewall between the trusted domain and the security perimeter domain to enforce the access policies to the trusted domain. We also need a network IDS to monitor the network to detect possible attacks.

  - Since the main container and the database are in this domain, if there is a successful attack in this domain, the whole system security will be jeopardized. For the operation system protection of this domain, we need not only the protections adopted for the operation system in the security perimeter domain, such as antivirus and security patches, but also host IDSs to strengthen the security. A host IDS is usually installed on individual workstations or servers and can be used to monitor inappropriate or anomalous activities and insider attacks.

In this e-hospital, the host IDS should be installed on database servers and on the machine where the main container resides. We can set host IDS to monitor the modification of system files, reconfiguration of important settings and manipulation of audit logs to detect possible attacks or unintentional mistakes.

- Authentication and message encryption are the same as those in the other two domains. We use private keys generated from the identity of the agent or the container as authentication. We use keys generated from an identity based two party key agreement protocol to encrypt the messages exchanged.

- The database is in the trusted domain and contains all the private information of patients. The database has its own protection mechanism to prevent unauthorized access and guarantee data integrity. However, the protection mechanism is not enough for patients' privacy protection. The original data are kept in clear text in the database and the private information can still be viewed or disclosed by people without need-to-know privilege such as database administrators or attackers who take advantage of weak security configurations of the database. We plan to encrypt the original data before they are stored in the database. The details can be found in the following section. The key used to encrypt the data is a symmetric key. When data are requested by end users, they are decrypted first and then protected by another symmetric key which is protected by the corresponding policy and finally sent to end users.

- Accountability plays an important role in the trusted domain as well. As in the security perimeter domain, we keep the system log and application audit log, and set the host IDS to monitor these logs to detect possible attacks or violations.
- All the equipment in the trusted domain should be put in the server room as those in the security perimeter domain and monitored by the hospital surveillance system.
- As we have noticed, all the most secure information is contained in the trusted domain and usually the maintenance is done by the system administrator or the network administrator. In order to reduce the possibilities of fraud, sabotage, and misuse of information, we need a security administrator to perform security related tasks. For example in the database maintenance, keys are maintained by the security administrator and routine backups are performed by the database administrator. Separation of duties makes sure one individual cannot complete a risky task by himself and thus increases the system security.

In this section, we have introduced our layer domain protection mechanism for this e-hospital. Our proposal is to use cryptography to implement authentication, data integrity, data encryption, secure communication channel, non-repudiation and database protection. We also need other protection mechanisms working in other security levels such as firewall, IDS, antivirus software and CCTV. Through layered protection, we can see that even if one protection mechanism fails we still have other protection mechanisms working in other levels to protect the system and thus we can provide stronger security architecture for this e-hospital system. We plan to provide security services in different domains to implement

cryptographic tasks such as key generation, encryption and decryption. The whole system architecture is illustrated in figure 3-2.

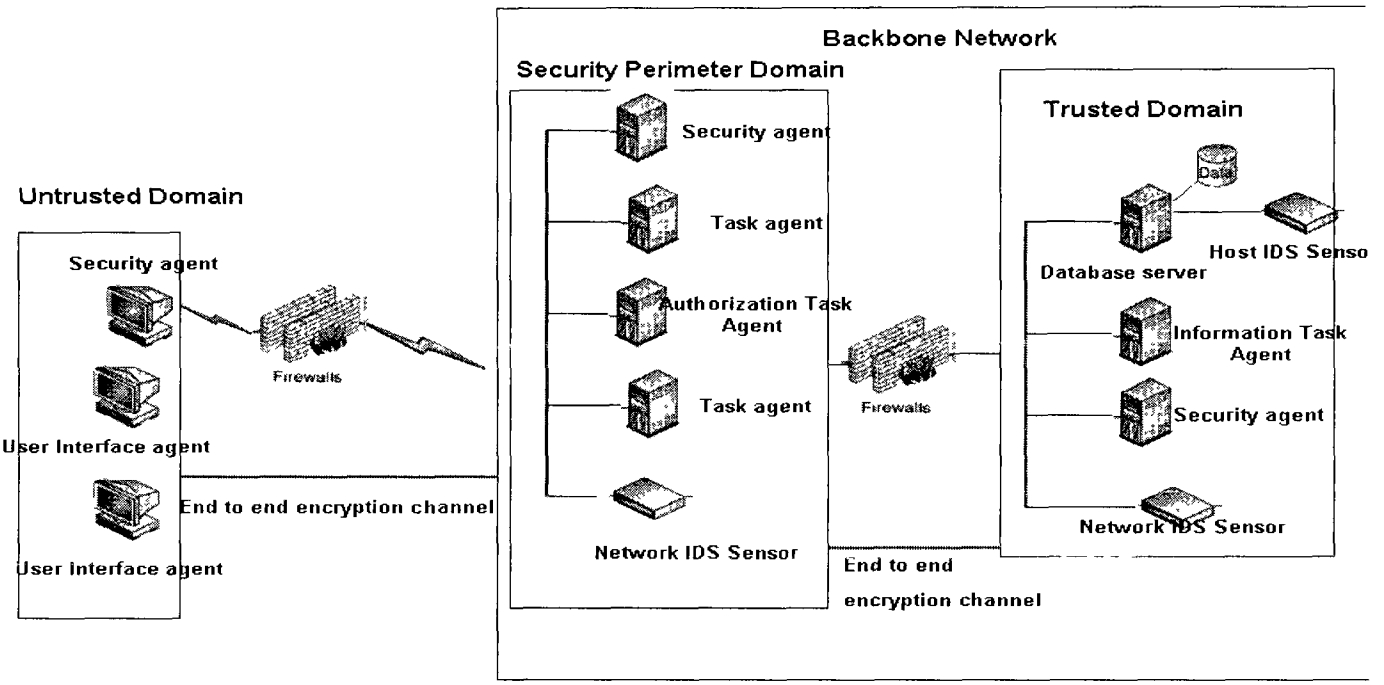

Figure3- 2 E-Hospital Security Architecture

3.3.2 Identity-Based Two Party Key Agreement

We use identity-based two party key agreement protocols in the computation of the session key. The session key is a symmetric key and is used to encrypt the messages exchanged between containers or agents. The public key of the agent or the container is its name which is its identity. If the session is over or times out, the session key and all of the information protected by this session key will be erased, which guarantees no medical data can be used outside of the hospital. The algorithm we use to compute the session key is similar to section 4 in [11]. The algorithm we use to encrypt or decrypt the real data is the symmetric algorithm

AES[8] The algorithm of the two party key agreement protocol and how it works are explained below

1 System set up Parameters

In this stage, parameters of the elliptic curve and master key pair of the TA[9] are generated The message space is $\{0,1\}^n$ The parameters Params=<q, G_1, G_2, e, l, P,H_1, H_2, s, P_S > where

- G_1 is the cyclic additive group of points on an elliptic curve defined over a prime field F_q,
- G_2 is the cyclic multiplicative group of points on an elliptic curve defined over field F_q^2,
- q is the large prime over which the elliptic curve is defined,
- P is the generator of G_1,
- e is a bilinear mapping function e $G_1 \times G_1 \rightarrow G_2$,
- l is the order of G1 and G2,
- H_1 is a hash function H_1 $\{0,1\}^n \rightarrow G_1$,
- H_2 is a hash function H_2 $G_2 \rightarrow \{0,1\}^n$
- s is the master key and a random number where $s \in Zq$,
- Ps is the public key of the TA and Ps=s*P,

2 Key Pair Generation

[8] AES Advanced Encryption Standard (AES) is a symmetric standard adopted by the U S Government
[9] TA Trusted Authority (TA) in identity based cryptography

Each agent's or container's public key Q is its name $Q_i=H_1(ID_i)$ where ID_i is the name of the agent or container. The corresponding private key S_i ($S_i=s*Q_i$) of the agent or the container is derived from its public key where s is the TA's private key.

3. Secure communication channel establishment:

In this step the two communicating parties use identity-based cryptography to compute a shared secret as their session key and use this session key to encrypt the messages exchanged between them. They compute the session key in the following steps:

- Party A chooses a random number $a \in Zq$ and sends Wa=a*Qa to agent B where Qa is A's public key;
- Party B chooses a random number $b \in Zq$ and sends Wb=b*Qb to agent B where Qb is B's public key;
- Party A computes Kab=e(Sa, Wb+a*Qb) where Sa is agent A's private key and Sa=s*Qa;
- Party B computes Kba=e(Sb, Wa+b*Qa) where Sb is agent B's private key and Sb=s*Qb;

Since

$$Kab$$

$$=e(Sa,Wb+a*Qb)$$

$$=e(s*Qa,b*Qb+a*Qb)$$

$$=e(s*Qa,Qb*(a+b))$$

$$=e(Qa,Qb)^{s(a+b)}$$

And

$$Kba$$

$$=e(Sb, Wa+b*Qa)$$

$$=e(s*Qb, a*Qa+b*Qa)$$

$$=e(s*Qb, Qa*(a+b))$$

$$=e(Qa,Qb)^{s(a+b)}$$

Therefore

$$Kab=Kba$$

The session key $K=H_2(Kab)$ can be used as a shared secret to encrypt messages between Party A and Party B

4 Secure communication

The session key is used to encrypt the communication between two parties in the e-hospital The algorithm used is the advanced encryption standard (AES) which is a symmetric algorithm and is a publicly accessible and open cipher approved by the NSA[10] for encrypting sensitive information The AES is a relatively new standard decided by the National Institute of Standards and Technology (NIST) to replace DES after the Data Encryption Standard (DES) was found too weak because of its small key size There were

[10] NSA The National Security Agency is a cryptologic intelligence agency of the United States government, administered as part of the United States Department of Defense

five finalists in the AES contest, and Rijndael, developed by Joan Daemen and Vincent Rijmen, was chosen by NIST to replace DES.

The AES is an iterated block cipher which supports a fixed block size of 128 bits and key sizes of 128,192 and 256 bits. The rounds of encryption depend on the key size. A key of 128 bits has 10 rounds, a key of 192 bits has 12 rounds, and a key of 256 bits has 14 rounds. The intermediate result of each round is called state. The state is a 4*4 rectangular array of bytes since the block size is 128 bits which is 16 bytes. The cipher key is pictured as a rectangular array with four rows and N_k columns where N_k=(the bit length of the key size)/32. During each round, the following operations are applied to the state:

1. SubBytes: every byte in the state is replaced by another one, using the Rijndael S-Box;
2. ShiftRows: every row in the 4*4 array is shifted a certain amount to the left;
3. MixColumns: a linear transformation on the columns of the state;
4. AddRoundKey: each byte of the state is combined with a round key, which is a different key for each round and derived from the Rijndael key schedule.

In the final round, the AddRoundKey operation is omitted. The AES standard has been incorporated into a lot of software technologies such as Microsoft .Net technology and several Java Technologies.

5. Session over:

In this stage, the session key and the information protected by the session key will be erased from both communicating parties. The deleting mechanism provided by most software

or operation systems doesn't really delete the data. It just removes the file's entry from file systems or erases the pointer to the memory where the data are. Therefore the sensitive data still can be recovered by operation system utilities or special software. In order to decrease the risk of unauthorized access of sensitive data from the media (in this case the hard disk), we need to securely erase the sensitive data.

There are two ways to perform secure deletion. One, overwrite the data or destroy the physical media. Two, encrypt the data and use the first methods to securely delete the key. The National Institute of Standards and Technology (NIST) has published a guideline for media sanitization [26]. NIST mentions four types of methods of sanitization for different types of media. Cleaning is suitable in the case of this e-hospital. Overwriting is one example of the cleaning methods and is supported by most high level programming languages. The aim of cleaning is to make sure the sensitive data cannot be retrieved by data, disk or file recovery utilities.

The goal of overwriting is to replace the written data with random data. NIST points out in its guideline that most media can be cleared effectively just by one overwrite. In order to counter more advanced data recovery techniques, specific overwrite patterns such as writing repeated alternating patterns of 1s and 0s several times should be used.

In the case of this e-hospital, we only need to securely erase the session key since the real data are protected by this session key and therefore we don't need to securely erase encrypted data. Through securely erasing the session key, we also decrease the risk that the private information is used outside of the hospital.

3.3.3 Access Control by Policy Based Cryptography

In this e-hospital, the end users access the patients' information according to the hospital's policy. As mentioned in the previous sections, the access policy is complicated and changes with time or job rotation. Also, in a distributed agent system, traditional access control is not practical. Bagga and Molva's policy based cryptography scheme [4] provides a possible solution to enforce end users' access control in this e-hospital. However, their scheme cannot solve the security objective of non-repudiation. We extend their scheme to address the non-repudiation problem. Our scheme also borrows ideas from certificateless public key cryptography[1].

3.3.3.1 Original Policy Based Cryptography Scheme

The policy-based cryptography scheme by Bagga and Molva is based on the idea that those who have the private keys of the corresponding policy are regarded as legal users and can decrypt the message protected by the policy.

In Bagga & Molva's policy based cryptography, the access policy is presented by conjunctions ($\wedge$) and disjunctions ($\vee$) of conditions. Each condition is defined through a pair <TA,A> where A is an assertion and TA is a trusted authority who is responsible for checking and certifying A's validity and issuing credentials corresponding to valid assertions. An assertion can be the information about the subject's attributes, properties, capabilities, etc. For each condition <TA,A>, the credential denoted $\varsigma(R,A)$ can be generated only by the TA using its secret master-key s and the validity of the credential can be checked using the TA's public key R.

The user has been issued the credential $\varsigma(R,A)$ if and only if he fulfils the condition <TA,A>. Let the expression 'user$\leftarrow \varsigma(R,A)$' denote the fact the 'user' has been issued credential $\varsigma(R,A)$ and let the expression 'user$\leftrightarrow$<TA,A>' denote the fact that the 'user' fulfils the condition <TA,A>. Then we get the following property: user$\leftrightarrow$<TA,A>$\Leftrightarrow$ user$\leftarrow\varsigma(R,A)$. A policy denoted as 'pol' is written in conjunction-disjunction normal form: $pol=\wedge_{i=1}^{m}[\vee_{j=1}^{mi}[\wedge_{k=1}^{mi,J} <TA_{i,j,k}, A_{i,j,k}>]]$.

If a user fulfils a policy, he has been issued all the corresponding credentials of the conditions in the policy. Let 'user$\Leftrightarrow$pol', for $pol=\wedge_{i=1}^{m}[\vee_{j=1}^{mi}[\wedge_{k=1}^{mi,J} <TA_{i,j,k}, A_{i,j,k}>]]$, denote the fact that user meets the policy and let the expression 'user$\leftarrow\varsigma(R,Ai,j,k)$' denote the fact that the user has been issued all the credentials included in the policy. Then the following property holds. user$\Leftrightarrow$pol $\Leftrightarrow\forall_i\in\{1,\ldots,m\},\exists_{ji}\in\{1,\ldots,m_i\}$: user$\leftarrow\varsigma(R,A_{i,ji})$. From the description we can see that if the user has been issued all the corresponding credentials in a policy then he is supposed to meet the policy. In this way, policy based cryptography can be used in access control.

The algorithm used in policy based cryptography is similar to the identity-based cryptography described above. The parameter setup and key pair generation are almost the same while in policy based cryptography the public key of a condition is $H_1(A)$ where A is an assertion and its corresponding private key is $s*H_1(A)$ where s is the master key of TA. The difference is the encryption and decryption procedures. Each conjunction of conditions is associated with a kind of mask denoted $\mu_{i,j}=H_2(g^r_{i,j}\|i\|j)$ where $g_{ij}^r=e(P_s,H_1(A_{i,j,k}))$. For each index i, a randomly chosen key t_i is associated with the disjunction $\vee_{j=1}^{ai}[\wedge_{k=1}^{\mu i,J} <TA_{i,j,k},$

$A_{i,j,k}$>]. Each t_i is encrypted a_i times using each mask $\mu_{i,j}$ and thus the end user can decrypt if he has one of the credentials in the disjunction conditions.

3.3.3.2 Original Certificateless Public Key Cryptography

Al-Riyami and Paterson proposed a public key cryptography scheme which they named certificateless public key cryptography (CK-PKC).In contrast to traditional public key cryptography, CL-PKC doesn't require the use of certificates to guarantee the authenticity of the public key. It relies on the TA who possesses a master key. In this respect, CL-PKC is similar to identity based cryptography (IBE). However, CL-PKC does not only rely on the TA, but also relies on the contribution of the secret known to the end user only to generate the final key pair. Therefore CL-PKC does not suffer from the key escrow property in IBE. Also CL-PKC addresses the non-repudiation problem in IBE since the final key pair is generated by the combination of the master key of the TA and the user secret key. CL-PKC can be seen as a scheme that is intermediate between traditional certificated public key cryptography and IBE.

CL-PKC uses a pairing based cryptography. It extends the IBE scheme in the key pair generation. All the parameters we use to explain CL-PKC are the same as the identity based two party key agreement protocol in 3.2.2. The end user's final private key is generated in two stages. First, an identity based partial private key $s*H_1(ID_i)$is generated where ID_i is the end user's identity. Second, the user chooses his secret x_i where $x_i \in Zp$ and generates his final private key $S_i= x_i *s*H_1(ID_i)$. The corresponding key pair is $<X_i,Y_i>$ where $X_i= x_i *P$, $Y_i== x_i *s*P$ where P is the generator of the group G1.

The basic scheme of CL-PKC works as follows. To encrypt a message M for the user, an entity first checks if the equation $e(X_i, P_s)=e(Y_i,P)$, where Ps is the public key of TA, holds. If it holds, a random number t from Zp is chosen and the ciphertext $C=<t*P, M\oplus H_2(e(Q_i,Y_i)^t)>$ where $Q_i = H_1(ID_i)$, is set to the recipient.

It is easy to see that to decrypt C=<U,V>, the user can use his private key $S_i= x_i *s*H_1(ID_i)= x_i *s* Q_i$ and compute

$$V\oplus H_2(e(S_i,U))$$

$$= M\oplus H_2(e(Q_i,Y_i)^t\oplus H_2(e(S_i,U))$$

$$=M\oplus H_2(e(Q_i,Y_i)^t\oplus H_2(e(x_i *s* Q_i, ,tP))$$

$$= M\oplus H_2(e(Q_i,Y_i)^t\oplus H_2(e(Q_i, P)^{xi*s*t})$$

$$= M\oplus H_2(e(Q_i,Y_i)^t\oplus H_2(e(Q_i, x_i*s*P)^t)$$

$$=M\oplus H_2(e(Q_i,Y_i)^t\oplus H_2(e(Q_i,Y_i)^t$$

$$=M.$$

Al-Riyami and Paterson also proposed a full scheme of CL-PKC which is chosen ciphertext attack secure. They considered two types of adversaries. Type1 adversary can replace public key with a key of its choice but has no access to the master key of the TA. Type2 adversary has access to the master key of the TA but can not replace the public key. They showed their scheme is chosen ciphertext attack secure under those two types of adversaries. The details can be found in [1].

3.3.3.3 Extended Policy Based Cryptography-Basic Scheme

In this section, we give a description of our proposed cryptographic scheme. Our scheme is based on policy based cryptography but we extend the scheme by adding a secret which is

only known to the end user. The final private key in our scheme is composed of two parts: the secret only known to end user and the private keys of the corresponding policy for the end user. The end users' public key is not only derived from the corresponding policy but also from their secret. In this way, we can provide non-repudiation in our scheme since only end users know their secret. Our scheme is a public key scheme and the real message is protected by a symmetric key which is protected by the end users' public key. We also propose a way to address the key revocation problem.

After the end user authenticates himself to the system, the private keys of the corresponding policy for the end user are distributed to him over a secure communication channel mentioned in section 3.2.2. The end user uses his secret plus the private keys received to generate a final private/public key pair and publishes his public key. When the end user requests the data from the system, the system encrypts the data with end user's public key and sends the encrypted the information to the end user. Only those who know the secret and have the private keys of the corresponding policy can decrypt and retrieve the message. When the communication session is over, the session key used to protect the real message is erased from the end user's machine securely.

In our scheme, the policy is composed of conjunctions of conditions only, since our aim is to gain the security objectives of need-to-know and the least privilege. For example one of the policies is "doctors from department1 can access patients' medical history between 9am and 5 pm". We can break this policy into three conditions. Condition1 is the assertion A_1"doctors from dedpartment1"; Condition2 is the assertion A_2 "patients' medical history"; and condition 3 is the assertion A_3 "between 9am and 5pm". The policy Pol can be represented by Pol=<TA,A_1>&<TA,A_2>&<TA,A_3>. The corresponding credentials K for

this policy are: $K_1=s*H_1(A_1)$, $K_2=s*H_1(A_2)$ and $K_3=s*H_1(A_3)$ where s is the master key of the TA.

The credentials above are just partial keys for those conditions. To address non-repudiation, we need a secret known only to the end user to construct the final private keys. This secret s_{user} is the end users' private key. The final credential for one condition is $s_{user}*K$ where K is the credential generated from policy based cryptography. Using the example above, the end user A chooses a secret s_a as its private key and the final private keys for this policy are $K_{1a}=s_a*K_1=s_a*s*H_1(A_1)$, $K_{2a}=s_a*s*H_1(A_2)$ and $K_{3a}=s_a*H_1(A_3)$. Legal users must have those three keys to decrypt. The public keys used to encrypt under this policy are generated by the end user's secret and the TA's public key.

Suppose each policy $pol=\wedge_{i=1}^{m} <TA,A_i>$ where A_i is an assertion for one condition and its validity can be checked by TA. The algorithm used in our basic scheme is detailed below:

- System set up: Parameters

The parameter setup is the same as that in 3.2.2. The message space is $M=\{0,1\}^n$ and the cipertext space is $C=G_1*\{0,1\}^n$ where n is the bit-length of the plaintext.

- Partial Private key generation

In this step, partial private keys for the policy are generated. The private keys for this policy are $\wedge_{i=1}^{m} s*H_1(A_i)$, where s is the master key of the TA and A_i is the assertion of each condition i. As you may have noticed, $H_1(A_i)$ is the public key in identity based cryptography.

- Private key generation

In this step, the final private keys for the policy are generated. The end user selects randomly $s_a \in Z_q^*$. The algorithm takes s_a and the partial private keys of the policy and generates the private keys for the policy $P_{sa}=\wedge_{i=1}^{m} s_a*s*H_1(A_i)$.

• Public Key Generation

At this point, the public key used to encrypt under the policy has been generated. The public key $P_{pol}=<X_{pol},Y_{pol}>$ where $X_{pol}=s_a*P$, $Ypol=s_a*Ps$. P is the generator of G_1 and Ps is the public key of the TA. $Ps=s*P$ where s is the master key of the TA.

• Encryption

In this step, the message M is encrypted by the conditions of the policy and the public key of the policy $P_{pol}=<X_{pol},Y_{pol}>$. The following steps are performed:

- Check if $X_{pol},Y_{pol} \in G1$ and if $e(X_{pol}, Ps)= = e(Y_{pol}, P)$. If it doesn't hold, output $\perp$ and abort the encryption;
- Choose randomly $r \in Z_q^*$, compute and output the cipertext $C=<rP, M \oplus H_2(\prod_{i=1}^{m} e(H1(Ai), Ypol)^r)>$.

• Decryption

In this step, the legal users decrypt the message by the private key of the policy. Suppose the cipertext is C=<U,V>; the legal user computes

$$V \oplus H_2(e(P_{sa},U))$$

$$=M \oplus H_2(\prod_{i=1}^{m} e(H1(Ai),s_a*s*P)^t) \oplus H_2(e(\sum_{i=1}^{m} s_a*s*H_1(A_i),rP))$$

$$=M \oplus H_2(\prod_{i=1}^{m} e(H1(Ai),s_a*s*P)^t) \oplus H_2(\prod_{i=1}^{m} e(s_a*s*H_1(A_i),rP)$$

$$=M \oplus H_2(\prod_{i=1}^{m} e(H1(Ai),s_a*s*P)^t) \oplus H_2((\prod_{i=1}^{m} e(H_1(A_i), s_a*s*P)^t)$$

$$=M.$$

3.3.3.4 Extended Policy Based Cryptography-Full Scheme

In this section, we extend our basic scheme to a full scheme which is adaptive chosen ciphertext attack (IND-CCA2) secure. IND-CCA2 is the standard acceptable notion of security for a public key encryption scheme. An encryption scheme is IND-CCA2 secure if no polynomially bounded adversary A has a non-negligible advantage against the Challenger C in the following 3-phase game. In phase 1, A may make decryption queries on ciphertexts of its choice. In the challenge phase, A chooses two messages M0, M1 and is given a challenge ciphertext C* for one of these two messages (i.e., the Challenger randomly chooses b$\in$ {0,1} and encrypts message Mb). In phase 3, A may make other decryption queries but not on C*. The game ends with A's guess b' on the value of b (i.e, A's guess as to which message was encrypted to C*). The encryption scheme is IND-CCA2 secure if the adversary's advantage Adv(A)=(Pr[b'=b]-1/2) is negligible.

Below is the algorithm of our full scheme, the proof of why our full scheme is IND-CCA2 secure is given at the end of Section 3.4.

- System setup: Parameters

  This step is similar to the setup in the basic scheme except that we add two cryptographic hash functions H_3: $\{0,1\}^n \times \{0,1\}^n \rightarrow Z_q^*$ and H_4: $\{0,1\}^n \rightarrow \{0,1\}^n$. The message space is the same as in the basic scheme but the ciphertext space has been changed to $G_1 * \{0,1\}^{2n}$.

- The partial private key generation, private key generation and the public key generation are the same as in the basic scheme.

- Encryption

To encrypt the message M under the policy and its public key $P_{pol}=<X_{pol},Y_{pol}>$, the sender needs to do the following steps:

1) Check if $X_{pol},Y_{pol} \in G1$ and if $e(X_{pol}, Ps)= e(Y_{pol}, P)$. If it doesn't hold, output $\perp$ and abort the encryption;
2) Choose a random $\sigma \in \{0,1\}^n$ and set $r=H_3(\sigma, M)$;
3) Compute and output the ciphertext $C= <rP, \sigma \oplus H_2(\prod_{i=1}^{m} e(H_1(A_i), Y_{pol})^r), M \oplus H4(\sigma)>$;

• Decryption

In this step, the legal user decrypts the message using the private key of the policy.

Suppose the ciphertext $C=<U,V, W>$, the legal user needs to performs the steps below:

1) Compute $V \oplus H_2(e(P_{sa},U))= \sigma'$;
2) Compute $W \oplus H_4(\sigma')=M'$;
3) Set $r'=H_3(\sigma',M')$
4) Test if $U=r'P$. If not output $\perp$ and reject C, otherwise output M' as the decryption of C.

The decryption computation uses the fact that for legal users:

$$H_2(\prod_{i=1}^{m} e(H_1(A_i), Y_{pol})^r) = H_2(e(P_{sa},U)).$$

Since

$$H_2(\prod_{i=1}^{m} e(H_1(A_i), Y_{pol})^r)$$

$$= H_2(\prod_{i=1}^{m} e(H_1(A_i), s_a*s*P)^t)$$

$$= H_2((\prod_{i=1}^{m} e(s_a*s*H_1(A_i), rP)$$

And

$$H_2(e(P_{sa},U))$$

$$=H_2(e(\Sigma_{i=1}^{m}(s_a*s*H_1(A_i),rP)$$

$$=H_2(\prod_{i=1}^{m}e(s_a*s*H_1(A_i), r*P).$$

Therefore

$$H_2(\prod_{i=1}^{m}e(H_1(A_i), Y_{pol})^r) =H_2(e(P_{sa},U)).$$

3.3.4 Database Security

In the database protection, we plan to encrypt the database contents at rest in the database. We use symmetric key cryptography to perform real-time I/O encryption and decryption of the data and log files. The encryption uses a database encryption key (DEK). The DEK is a symmetric key secured by the master key of the database.

Encryption of the database file is performed at the page level. The pages in an encrypted database are encrypted before they are written to disk and decrypted when read into memory. Thus this symmetric key cryptography is totally transparent to the application system and no changes are needed in the application system.

The symmetric key is loaded into the memory when encrypting or decrypting data. To ensure the security of this symmetric key, we need to securely erase the symmetric key after encryption or decryption. The secure erasing method we use is the same as we use in 3.3.2 to erase the session key. We replace the symmetric key in the memory after we use it, with overwritten pattern of alternating 1s and 0s several times.

The symmetric key can be placed in the hard disk since it is protected by the master key of the database. The master key should not be placed in hard disk. It should be placed in

other media attached to the database server and only can be accessed by the system service when the database starts up. The master key should be protected by other methods such as passwords only known to the security administrator.

To avoid unauthorized access of those keys, we can set the host IDS to monitor the database log file to trace possible attacks.

Symmetric key cryptography is very suitable to protect data from intruders and insiders. If the DBA has no privilege to access the original data, he/she cannot access the data. In order to decrease the risk of unauthorized access by the DBA, we should adopt separation of duty constraints: key generation and backup should be done by a security administrator other than the DBA.

Encryption of the database at the rest is supported by many database vendors, such as Microsoft and Oracle. Their schemes are more or less the same as we describe above.

3.4 Security Analysis

In this section, we analyze how our design meets the security objectives of this e-hospital. We also give a description of the security of our full scheme. How our design meets the security objectives of this e-hospital are listed below:

- Sensitive information should be protected with a higher level of security.

  We design different protection mechanisms to protect the sensitive information. When data are stored in the database, we encrypt all the data to make sure they are secure. When end users request sensitive information, we use our extended policy based cryptography scheme to control end user access. When data are in transmission, we

provide a secure communication channel to protect the data. Through layered cryptographic protection, we guarantee the confidentiality of the sensitive information.

- Patients' records should be returned to the hospital at the end of the day.

Our design meets this objective in two ways. First we monitor the session, and when the session ends, all the data stored temporarily on end users' devices are erased. Secondly we use one condition as time control in the end user access policy. For example we define one condition as from 9am to 5pm and encrypt the message under this condition. The details of how the encryption scheme works can be found in 3.2.3.2.

- According to the hospital security policy, no sensitive data is allowed to be used outside of the hospital.

This objective can be achieved by the design mentioned in "Patients' records should be returned to the hospital at the end of the day". A session ends when an end user device is taken outside of the hospital (e.g., beyond the range of a wireless base station), and all the data in that end user device is erased.

- The confidentiality and integrity of the sensitive information must be guaranteed to make sure an unauthorized party can't view and modify protected information.

To ensure data confidentiality and integrity, we use cryptography. We use symmetric cryptography to encrypt the sensitive information in the database. When the sensitive information leaves the database, we encrypt the communication channel to make sure sensitive information is encrypted during transmission. We also use our policy based scheme to control end users' access and provide the confidentiality for the sensitive data. We use a one way hash function to generate a message digest to ensure data integrity.

The message digest is transmitted with the message and protected by the session key to make sure there is no unauthorized access.

- Non-repudiation of the end user

This objective is achieved through our extended policy based cryptography scheme and the application audit log. The private key of the policy is composed from the master key of the TA and a secret only known to the end user. The application audit log keeps track of the access of the message. Therefore if a message is read by an end user, he cannot deny since he is the only one who knows the corresponding secret. In this way, the recipient of the message can be confirmed.

- End users must authenticate to the system before they login to the e-health system.

End users access this e-hospital through two factor authentication: username/password and their RFID card.

- In order to prevent malicious agents attacking other agents, agents should authenticate to each other before data exchange;

In this e-hospital, each agent has a unique identity. Each agent has a private key generated from its identity and agents use their private key to authenticate to each other.

- A normal container and the main container should authenticate each other before the normal container registers to the main container. An illegal normal container is not allowed to register to the main container and the illegal main container is not allowed to provide service to other normal containers;

This objective can be achieved by our identity based cryptography mentioned above.

- Access should be limited to a need-to-know basis and also access should be based on the fact that individually identifiable health information is collected, accessed, used or disclosed only with a patient's consent;

This objective is achieved by our extend policy based cryptographic scheme. We define access policy as the conjunction of conditions. The message is protected by the policy and only the legal user can have the corresponding decryption keys of the policy. The patients can give their consent by signing the related file. After that, we can encode the medical data with patients' consent into one condition; only those medical staff who have the corresponding private key for this condition can access the patients' records.

- Sensitive data in the database should be protected against unauthorized viewing, copying, use or modification;

We use symmetric encryption to protect the data in the database. We also separate the duties of the security administrator from that of the DBA to decrease the risk of unauthorized access. End users' access is controlled by our extended policy based cryptography scheme.

- The physical environment of this e-hospital should be protected from attacks.

The devices in the backbone system are placed in the server room and monitored by CCTV. The server room is protected by locks and can only be accessed by the system administrator.

- The communication between the e-hospital and other hospitals should be protected against malicious attacks such as network sniffers.

All the data transmitted between this e-hospital and other hospitals are encrypted. The encryption and decryption is done by the hospital gateway between this e-hospital and other hospitals. This e-hospital just accepts or renders the medical data to the hospital gateway.

- The system should keep the audit trails to identify security relevant events such as possible attacks or unauthorized access.

In this e-hospital, we keep system log and application audit logs and set the host IDS to monitor those logs to detect possible attacks or violations.

- The communication channel between agents should be protected against malicious attacks.

All the communication in this e-hospital is protected by encryption. We use identity based two party key agreement protocols to distribute the encryption key and encrypt the message transmitted.

- Different agents in this e-hospital are responsible for different tasks and deal with different data with different security levels. Therefore they should be in different networks and those different networks should be separated by protection mechanisms.

We define three domains according to the security levels in this e-hospital. We use firewalls to separate different domains into different networks. For each domain we provide different protection mechanisms such as network intrusion detection system, host-based intrusion detection system and cryptography.

- The operation system should be protected from malicious code.

We use the third party antivirus software to protect the operation system from virus and malicious code attacks. We also apply security patches for the operation system in time to decrease the risk that the attacker takes advantage of the vulnerability of the operation system. Finally we use a host-based IDS system to detect possible attacks.

- The continuity of operation of this e-hospital should be guaranteed. Authorized users are supposed to be able to use the e-hospital system even in the event of attacks.

To ensure the availability of the e-hospital system, we need to prevent a single point of failure in this e-hospital system. As we have noted, JADE relies on the main container to house the key platform services such as the AMS and DF and this makes the main container a potential single point of failure. The JADE platform provides the main container replication services to deploy a fault tolerant platform. Once the major main container fails, the stand-by main container can take over and can make sure that the business continues. We rely on the fault tolerance services provided by the JADE and ensure the availability of the e-hospital system.

In the above, we give a brief description of how our design meets the security objectives of this e-hospital. Now let's look at the security of our proposed cryptographic scheme. We mentioned adaptive chosen ciphertext attack secure (IND-CCA2) in section 3.2.3.2. In Identity Based Encryption scheme [7], they allow the adversary to extract private keys of his choice other than the ID being attacked. Their aim is to show that the IBE scheme is secure even if the attacker obtains some of the private keys of some IDs.

In our cryptographic scheme, we need to consider the adversary who can extract private keys of the policies of its choice since under some circumstances some of the private keys of

the policies might be obtained by the attacker. We also need to consider the situation where the adversary can replace the public key of the policy of his choice. The attacker can replace the public key of those policies whose private keys or partial private keys are available to him, or he can replace the public key of the policy which will be attacked by him. The worst case in our scheme is that the adversary has access to the master key. However, we need to constrain this adversary who can access the master key from replacing the public key of the policies of its choice.

We follow the security model in [1]and list two kinds of adversaries in our scheme:

- Type 1 adversary: such an adversary has no access to the master key but he can extract partial private keys, or private keys, or both, of the policies of his choice, replace public keys of his choice and make decryption queries. The constraints with the Type 1 adversary are as follows: the adversary cannot extract the private key for the policy challenged at any point; the adversary cannot extract the private key for the policy whose public key has already been replaced; the adversary cannot make decryption queries on the challenge ciphertext C^* encrypted with the public key of the challenged policy.
- Type2 adversary: the Type2 adversary has access to the master key. A Type2 adversary can compute partial private keys for itself, request public keys, make private key extraction queries and decryption queries of his choice. The constraints for type2 adversary are as follows: Type2 adversary cannot replace public keys at any point; Type2 adversary cannot extract the private key for the policy challenged at

any point; in phase two he cannot make a decryption query on the challenged ciphertext encrypted by the public key of the challenged policy.

Our cryptographic full scheme is IND-CCA2 secure if no polynomially bounded adversary of Type1 or Type2 has a non-negligible advantage against the challenger in the following game:

- System setup: The challenger takes a security parameter and generates the system parameters. The Type1 adversary gets system parameters except for the master key and the Type2 adversary gets the system parameters.
- Phase 1: The adversary issues a sequence of requests. Each request can be the partial private key extraction, the private key extraction, the public key replacement, the request for public key or the decryption query for the policy of his choice. Those requests are under the constraints mentioned above.
- Challenge Phase: Once the adversary decides that phase 1 is over, he outputs two messages M_0,M_1 and the policy challenged to the challenger. The challenger picks randomly one of the two messages (i.e., m_b, where b is 0 or 1), encrypts it under the policy challenged and its public key and delivers the ciphertext to the adversary. Again all constraints mentioned above apply. If the adversary has replaced the public key in the wrong form, the game ends immediately.
- Phase 2 : The adversary may issue requests as in the phase 1.

The game ends with the adversary's guess of b' for b. The adversary's advantage in this game is Adv(A) = Pr(b=b')-1/2. The adversary's advantage is negligible given the

assumption that there is no polynomially bounded algorithm that can solve the GBDHP in groups with non-negligible advantage.

Now we give a general explanation why our scheme is secure against those two types of adversaries. For the adversary who can replace the public key of its choice, our aim is to show the cryptographic scheme is secure against the adversary. If the adversary does not extract the partial private keys of the policy, we need to show the adversary cannot decrypt correctly the message protected by the policy. If the adversary extracts the partial keys of the users he wants to attack, our aim is to show the adversary cannot decrypt the message for those whose partial private key he does not know.

We have authentication mechanisms in the system. If the attacker wants to attack another user, he must first authenticate to the system. The following analysis is based on the assumption that the attacker has successfully fooled the authentication server and wants to publish a public key of his choice.

Suppose the adversary Bob wants to get the message intended for user Alice. Bob logs on to the system, publishes a public key $<X_{alice'}, Y_{alice'}>$ and claim this is Alice's public key. Here $X_{alice'}=s_a'*P$, $Y_{alice'}=s_a'*s*P$ where s_a' is chosen by Bob as a fake private key for Alice, P is the generator of the group G1, s is the master key of the TA and s*P is the public key of TA. In this case Bob does not extract the partial private keys of the policy for Alice. Alice's partial private key P_{salice} is denoted as:

$P_{salice}=\wedge_{i=1}^{m} s*H_1(C_{ialice})$ which means Alice's partial private key is composed of m conditions and each condition i corresponds to a private key $s*H_1(C_{ialice})$ where s is the master key and C_{ialice} is the condition of the policy for Alice.

Suppose the system encrypts a message under Alice's fake public key chosen by the adversary Bob. The encryption and decryption procedure is the same as listed in our full scheme. The ciphertext in this case is

$$C= <rP, \sigma \oplus H_2(\prod_{i=1}^{m} e(H_1(C_{ialice}), Y_{alice'})^r), M \oplus H4(\sigma)>.$$

As we have explained in our full scheme the decryption procedure uses the fact that:

$$H_2(\prod_{i=1}^{m} e(H_1(A_i), Y_{pol})^r) = H_2(e(P_{sa}, U)).$$

In the case of the adversary Bob attacking the user Alice, Bob computes $H_2(e(P_{salice'}, U))$ and if he can make $H_2(e(P_{salice'}, U)) = H_2(\prod_{i=1}^{m} e(H_1(C_{ialice}), Y_{alice'})^r)$ he can successfully attack the user Alice. The $P_{salice'}$ is the fake private key of Alice where $P_{salice'} = s_{a'} * \Sigma_{i=1}^{m} s * H_1(C_{ibob})$ since Bob doesn't know the partial private keys of the policy for Alice (therefore he will use his own or other partial private keys known to him to construct the fake private key for Alice).

Since

$$H_2(\prod_{i=1}^{m} e(H_1(C_{ialice}), Y_{alice'})^r)$$

$$= H_2(\prod_{i=1}^{m} e(H1(C_{ialice}), s_{a'} * s * P)^r)$$

$$= H_2((\prod_{i=1}^{m} e(s * H_1(C_{ialice}), s_{a'} * rP)$$

And

$$H_2(e(P_{salice'}, U))$$

$$= H_2(e(\Sigma_{i=1}^{m} (s_{a'} * s * H_1(C_{ibob}), rP)$$

$$= H_2(\prod_{i=1}^{m} e(s * H_1(C_{ibob}), s_{a'} * rP)$$

Therefore

$$H_2(e(P_{salice'}, U)) \neq H_2(\prod_{i=1}^{m} e(H_1(C_{ialice}), Y_{alice'})^r)$$

In order to make the explanation clear, we write the public key of the policy for Alice as $P_{alice}=\wedge_{i=1}^{m} H_1(A_i)=kP$ where $k\in Z_q^*$ and P is the generator of G_1. The reason the above equation holds is because for $\forall i$, $H_1(C_{ialice})$ is a point belonging to G1 and therefore $H_1(C_{ialice})$ can be represented by the generator P of G_1 as x_iP where $x_i\in Z_q^*$. We define $k\in Z_q^*$ and $k=\Sigma_{i=1}^{m} x_i$, and thus we can write $P_{palice}=kP$.

From the analysis above, we can see what Bob is trying to do is the following:
Given the generator P of the group G_1, rP (U), sP(master key) and the public key of the policy kP for the user Alice, Bob is trying to compute $e(P,P)^{ksr}$. This is equivalent to solving the GBDH[11] problem in the group G_2. As long as the GBDH problem is hard in G_2, Bob has negligible advantage to decrypt the message for Alice.

We give an analysis of the adversary who can replace the public key of its choice and who does not extract the partial private keys of the policy for the user under his attack. For the adversary who can replace the public key of his choice and who already knows the partial private key (the secrets) of some users, our aim is to show the adversary cannot use what he knows to attack other users whose partial private key he does not know. The analysis is the same as the above case of Bob attacking Alice.

For Type2 adversary who knows the master key but who cannot replace the public key, we need to show this kind of adversary cannot decrypt the message for the user under the attack. As described in the case of Bob attacking Alice, if the Type2 adversary can compute $H_2(\prod_{i=1}^{m} e(H_1(C_{ialice}), Y_{alice'})^r)$ or $H_2(e(P_{salice}, U))$ where $Y_{alice}=sa*s*P$,

[11] GBDH Problem: For randomly chosen $a,b,c\in z_q^*$, given $g, g^a, g^b, g^c \in G1$, compute $e(g,g)^{abc} \in G2$ with the help of a DBDH oracle which answers whether a given tuple is a BDH tuple or not.

$P_{alice}=\Sigma_{i=1}^{m}(s_a*s*H_1(C_{ialice}))$ and s_a is the partial private key (or secret) only known to Alice, then he can decrypt the message for Alice under attack. Either computation can be reduced to the following: given rP, Y_{alice} (s_a*s*P) and public key of the policy (KP) for Alice, compute $e(P,P)^{r*sa*s*k}$. As long as the GDBH problem is hard in G_2, Type2 adversary's advantage to successfully attack Alice is negligible.

3.5 Summary

This chapter presents the security architecture for this e-hospital. We follow the international security standard ISO17799 as our guide in the design of the security features for this e-hospital. The security objectives of this e-hospital are reached by complying with the patients' privacy protection regulations, risk assessments and end users' requirements. After we reach the security objectives of this e-hospital, we divide this e-hospital system into three domains according to their security levels. For each level we adopt different protection mechanisms to meet the security objectives. In the protection of the communication channel, we use an identity based two party key agreement protocol to distribute encryption keys. We extend policy based cryptography to control end users' access. Our proposed scheme can ensure data confidentiality, integrity and non-repudiation. We end the chapter by an analysis of how our design meets the security objectives of this e-hospital and we also discuss the security of our cryptographic scheme. In the next chapter, we will give the implementation of this e-hospital.

Chapter 4 System Implementation

In chapter3 we proposed a security architecture for the e-hospital. This chapter presents how to implement the security features for the e-hospital. We focus on the implementation of our proposed cryptographic scheme and the underlying mathematical computations. For other protection mechanisms, such as IDS system and antivirus, we only give general advice which can be found in chapter 3. This chapter is organized as follows. In section 4.1 we introduce the elliptic curve, the Tate Pair and the related mathematical background to compute the Tate Pairing. In section 4.2 we give the detailed algorithms used in our implementation. The algorithms include three parts: group computation on elliptic curves, the Tate pairing computation and the pairing-based cryptographic scheme implementation. In section 4.3, we give the test results using elliptic curves defined over prime numbers. We also test the encryption and decryption performance under different lengths of access policies. We end the chapter with a summary.

4.1 Introduction

In this section, we give an introduction to the mathematical backgrounds of elliptic curves, The Tate pairing and the mathematical background to compute the Tate pairing. The definitions and notation are from [23][33].

4.1.1 Definitions and Notation

1) Fields

Fields are abstractions of familiar number systems (such as the rational numbers Q, the real numbers R and the complex numbers C) and their essential properties. They consist of a set F with two operations, addition (denoted by +) and multiplication (denoted by *), that satisfy the following properties:

- (F, +) is an abelian group with identity denoted by 0.
- (F\{0}, *) is an abelian group with identity denoted by 1.
- The distribution law holds: (a+b)*c=a*c + b*c for all a,b,c $\in$ F.

If F is finite, then the field is said to be finite.

2) Field Operations

Addition and the multiplication are two basic operations on a field. Subtraction and division are defined in terms of addition and multiplication. For example, assume a,b $\in$ F, then a-b is defined as a+(-b) where –b is defined such that b+(-b)=0. –b is called the inverse of b in F. For a, b $\in$ F, a divided by b is defined as $a*b^{-1}$ where b^{-1} is the unique element in F such that $b^{-1}*b=1$. b^{-1} is called the inverse of b.

3) The Order and Characteristic of Fields

The order of a finite field is the number of elements in the filed. Suppose a field F is defined over q where $q=p^m$ and p is a prime. p is called the characteristic of F, and m is a positive integer and called the embedded degree of F. If m=1, then F is called a prime field. If m>=2, then F is called an extension field. For any prime power q, there is essentially only one finite field of order q which means any two finite fields of order q are strictly the same except the labelling of elements may be different.

4) Prime Fields

Finite fields F_p of order p (where p is a prime) are defined as consisting of the integers {0,1,2….,p-1} with addition and multiplication modulo p.

5) Binary Fields

Finite field F_2^m of order 2^m is called a binary field or characteristic-two finite field.

6) Subfields of a finite field

A finite field F_p^l of order p^l is said to be a subfield of the finite field F_p^m of order p^m if F_p^l is itself a finite field with respect to the operation of F_p^m. A finite field F_p^m has precisely one subfield of order p^l for each positive divisor l of m. The elements of F_p^l are the elements $a \in F_p^m$ and $a^{\wedge} p^l=a$.

7) Legendre symbol

A nonzero element $a \in Zn$ is called a quadratic residue modulo n if there exists some $x \in Zn$ such that $x^2=a(\text{mod } n)$. If no such x exists, a is called a quadratic nonresidue modulo n. Legendre symbols are used to indicate whether an integer is a quadratic residue or not. Let p be a prime and let a be an integer; the legendre symbol (a/p) is defined as follows:

- 0 if p divides a
- 1 if a is a quadratic residue modulo p
- -1 if a is a quadratic nonresidue modulo p

Let a and b be integers and let p and q be primes; the legendre symbols have the following properties:

- $(a/p)=a^{(p-1)/2} \text{ (mod p)}$
- (ab/p)=(a/p)(b/p)
- If a=b(mod p) then (a/p)=(b/p)
- $(2/p)=(-1)^{(p^{\wedge}2-1)/8}$

- $(p/q)=(q/p)(-1)^{(p-1)(q-1)/4}$

8) Divisor of a Rational Function f

A divisor of a rational function f is a way of characterizing f based on its zeros, where f(x)=0, and poles, where f(x)=+/-O (O denoted as infinity). The divisor of a rational function is denoted by div(f) and written as the sum of points where f has a zero or pole weighed by the multiplicities of the zeros and poles, with the convention that zeros get positive weights and poles get negative weights. For example, the function f(x) as defined below:

$$f(x)=(x-1)^2/(x+2)^3$$

has a zero of order 2 at x=1, a zero of order 1 at infinity and a pole of order 3 at x=-2. The divisor of f(x) is as follows:

$$div(f)=2(1)+(O)-3(-2)$$

In general, suppose f(x) is defined over S where $S=\{x_1,x_2,....x_i\}$ and $i\in Z$. f(x) can be defined as

$$f(x)=\prod(x-x_i)\wedge a^i$$

Then the divisor div(f) is written as

$$div(f)=\Sigma a_i(x_i)$$

From the above definitions, we have the following properties:

$$div(fg)=div(f)+div(g)$$

$$div(f/g)=div(f)-div(g)$$

9) Divisor on Elliptic Curve

Let E be an elliptic curve; a divisor on E is a formal sum of the form

$$D=\Sigma n_p(P)$$

where $P\in E$ and each n_p is an integer and all but finitely many n_p are zero.

10) Principal Divisor

Let E be an elliptic curve, a divisor D on E is said to be a principal divisor if there is a rational function f such that D=div(f) or D can be written as $D=\Sigma\ a_i(P_i)$ where $\Sigma ai=0$ and $\Sigma a_iP_i=O$. For example if P is a point on an elliptic curve of order n, then n(P)-n(O) is a principal divisor.

11) f(D)

If D is a divisor of the form

$$D=\Sigma a_i(P_i)$$

then a rational function f evaluated at D is defined as

$$f(D)=\prod f(P_i)\wedge a^i$$

12) Support of Divisors

Let D be a divisor of an elliptic curve in the form of:

$$D=\Sigma n_p(P)$$

then the support of D is the set of all points P where $n_p\neq 0$.

Let D_1 and D_2 be divisors of an elliptic curve. D_1 and D_2 are said to have disjoint support if the intersection of the support of D_1 and D_2 is empty.

13) Weil Reciprocity

Let f and g be rational functions defined over some fields. If div(f) and div(g) have disjoint support then f(div(g))=g(div(f)).

14) Equivalence of Divisors

Divisors D_1 and D_2 are equivalent if they differ only by a principal divisor. For example if $D_1=D_2+div(f)$ where div(f) is the divisor of the rational function f, then D1 and D2 are equivalent.

4.1.2 Elliptic Curve

An elliptic curve is generally defined by a Weierstrass equation [23] Let E be an elliptic curve defined over a field K, the Weierstrass equation is

$$E \quad y^2+a_1xy+a_3y=x^3+a_2x^2+a_4x+a_6$$

where a1,a2,a3,a4,a6 $\in$ K and the discriminant of E $\Delta\neq0$ $\Delta\neq0$ ensures that the elliptic curve is "smooth", that is, there are no points at which the curve has two or more distinct tangent lines The discriminant Δ is defined below

$$\Delta=-d_2^2d_8-8d_4^3-27d_6^2+9d_2d_4d_6$$

$$d_2=a_1^2+4a_2$$

$$d_4=2a_4+a_1a_3$$

$$d_6=a_3^2+4a_6$$

$$d_8=a_1^2a_6+4a_2a_6-a_1a_3a_4+a_2a_3^2-a_4^2$$

All the points (x,y) with (x,y)$\in$K which satisfy the equation above together with a point at infinity denoted as O construct the elliptic curve E defined over the field K By admissible changes of variables, a Weierstrass equation can be changed to simplified equations as follows

- If the characteristic of K is not equal to 2 or 3, then the simplified equation is

  $$y^2=x^3+ax+b \quad (4,1)$$

  where a,b$\in$ K and $\Delta =-16(4a^3+27b^2) \neq0$

- If the characteristic of K is equal to 2, then the simplified equation is

  $$y^2+xy=x^3+ax^2+b \quad (4\ 2)$$

  where a,b$\in$ K and $\Delta =b \neq0$

Or

$$y^2+cy=x^3+ax+b \quad (4,3)$$

where $a,b \in K$ and $\Delta = c^4 \neq 0$.

- If the characteristic of K is equal to3, then the simplified equation is:

$$y^2=x^3+ax^2+b \quad (4,4)$$

where $a,b \in K$ and $\Delta = -a^3b \neq 0$.

Or

$$y^2=x^3+ax+b \quad (4,5)$$

where $a,b \in K$ and $\Delta = -a^3 \neq 0$.

4.1.3 Group Law of Elliptic Curves

Elliptic curves used in cryptography are usually the curves defined by equation 4.1 or equation 4.2. Suppose E is an elliptic curve defined over the field K; all the points on E plus the point at infinity (which is the identity element on E), together with the addition operation on E, construct an abelian group. The elliptic curve cryptographic system can be constructed using this group. Let Char(K) denote the characteristic of K. Group laws for the elliptic curve prime field and the field of characteristic two are given blow.

1. Group law for E/K: $y^2=x^3+ax+b$ where $Char(K) \neq 2,3$ and a,b are integers.

   - $P+O=O+P=P$ for all $P \in E(K)$.
   - If $P=(x,y) \in E(K)$, then $(x,y)+(x,-y)=O$. The point$(x,-y)$ is called the negative of P and $-P \in E(K)$.
   - $O+O=O$.
   - Let $P=(x_1,y_1) \in E(K)$ and $Q=(x_2,y_2) \in E(K)$ where $x_1 \neq x_2$. Let $R=(x_3,y_3)=P+Q$, then

$$x_3=((y_2-y_1)/(x_2-x_1))^2-x_1-x_2.$$

$$y_3=((y_2-y_1)/(x_2-x_1))(x_1-x_3)-y_1.$$

The sum of P and Q can be explained in the following way: First draw a line through P and Q; the line intersects E at a third point. R is the reflection of this point about the x-axis. Figure 4-1 gives an illustration of how to add P and Q.

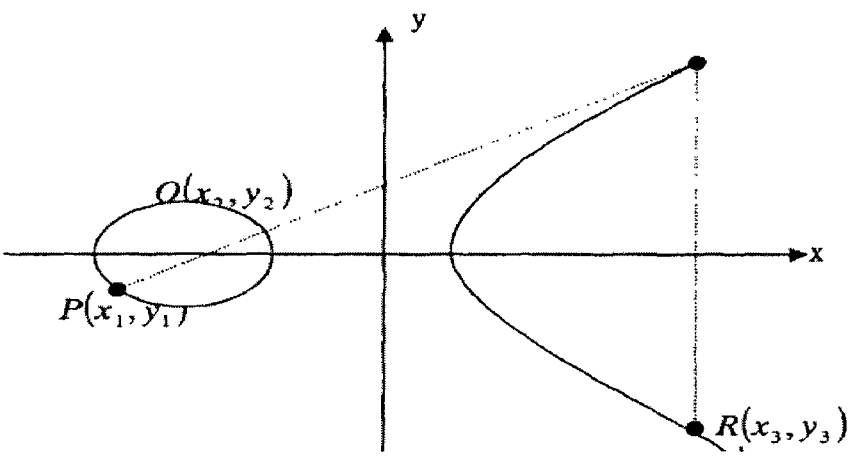

Figure4-1 Addition of P and Q.

- Let $P=(x_1,y_1) \in E(K)$, where $P \neq -P$. Let $L=2P=(x_3,y_3)$, then

$$x_3=((3x_1^2+a)/2y_1)^2-2x_1$$

$$y3=((3x_1^2+a)/2y_1)(x_1-x_3)-y_1.$$

The doubling of P is defined as follows: First draw a tangent line to the elliptic curve at P. This line intersects the elliptic curve at a second point. L is the reflection of this point about the x-axis. The illustration of the doubling of P can be found in figure 4-2.

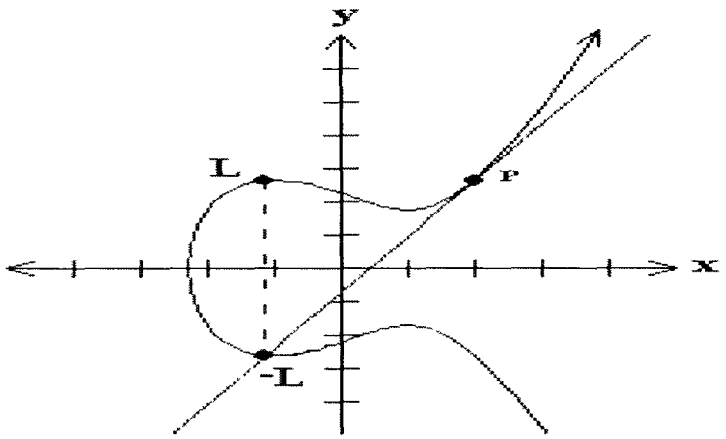

Figure4-2 Doubling of P

2. Group law for E/F_2^m: $y^2+xy=x^3+ax^2+b$

- $P+O=O+P=P$ for all $P \in E(F_2^m)$.
- Let $P=(x,y) \in E(F_2^m)$, $(x,y)+(x,x+y)=O$. The point $(x,x+y)$ is called the negative of P and denoted by $-P$. Also $-P \in E(F_2^m)$.
- $O+O=O$.
- Let $P=(x_1,y_1) \in E(F_2^m)$ and $Q=(x_2,y_2) \in E(F_2^m)$ where $x_1 \neq x_2$. Let $R=(x_3,y_3)=P+Q$, then

$$x_3=\lambda^2+\lambda+x_1+x_2+a$$

$$y_3=\lambda(x_1+x_3)+x_3+y_1.$$

with $\lambda=(y_1+y_2)/(x_1+x_2)$.

- Let $P=(x_1,y_1) \in E(K)$, where $P \neq -P$. Let $2P=(x_3,y_3)$, then

$$x_3=\lambda^2+\lambda+a=x_1^2+b/x_1^2$$

$$y_3=x_1^2+\lambda x_3+x_3$$

with $\lambda = (x_1+ y_1)/ x_1$.

ECC based cryptographic schemes rely on the scalar multiplication of elliptic curve points. Given an integer k and a point P with $P \in E(K)$, the scalar multiplication is the process of adding P to itself k times. Scalar multiplication can be computed by using adding rules together with doubling rules.

4.1.4 Tate Pairing

Identity-based cryptography can be constructed by Weil pairing or Tate Pairing. In practice, as observed in [17][39], the Tate Pairing can be computed more efficiently than the Weil Pairing. The Tate Pairing is used in our implementation. The definition of the Tate Paring is given below.

Let E be an elliptic curve over a finite field F_q where q is the characteristic of the field. The point at infinity on E is denoted by O. Let l be a positive integer which is coprime to q. Let k be a positive integer such that the field F_q^k contains the lth root of unity. Let $G=E(F_q^k)$, and define $G[l]=\{P \in G : [l]P=O\}$ and $lG=\{[l]P: P \in G\}$. Then G[l] is a group of exponent l. lG is a subgroup of G and the quotient group G/lG is a group of exponent l. The group G[l] and the quotient group G/lG are isomorphic which means they have the same number of elements, but the labeling of the elements may be different. The Tate Paring is a mapping $<P,Q>:G[l]*G/lG \rightarrow F_{q^k}^*/(F_{q^k}^*)^l$ where $F_{q^k}^* = F_q^k-\{O\}$.

The Tate Paring is defined as follows. Given points P and Q, find a function f such that $(f)= l(P)-l(O)$. Let D be a divisor equivalent to (Q)-(O) such that the support of D is disjoint from the support of (f). In most cases such a divisor can easily be constructed by choosing a random $S \in G$, $S \neq O$ or P, and defining D=(Q+S)-(S).[33]. The Tate Pairing of P and Q is defined to be $<P, Q>_l=f(D)$ where $f(D) =\prod_i f(P_i)^{n_i}$ and $D=\Sigma_i n_i P_i$ ($n_i \in Z$, $P_i \in G$). From the

definition above we can see that the Tate Pairing can be interpreted as f(D) which is an element of $F_q^{*k}/(F_q^{*k})^l$. The Tate Paring can be computed using Miller's algorithm which can be found in the following section.

Before explaining how Miller's algorithm works, we give an illustration of how to add two divisors on an elliptic curve. Suppose P_1,P_2 and P_3 are points on an elliptic curve E. Define line u to be the line which passes through P_1 and P_2. Define line v to be the vertical line which goes through P_3 where $P_3=P_1+P_2$. The illustration of u and v can be found in figure 4-3.

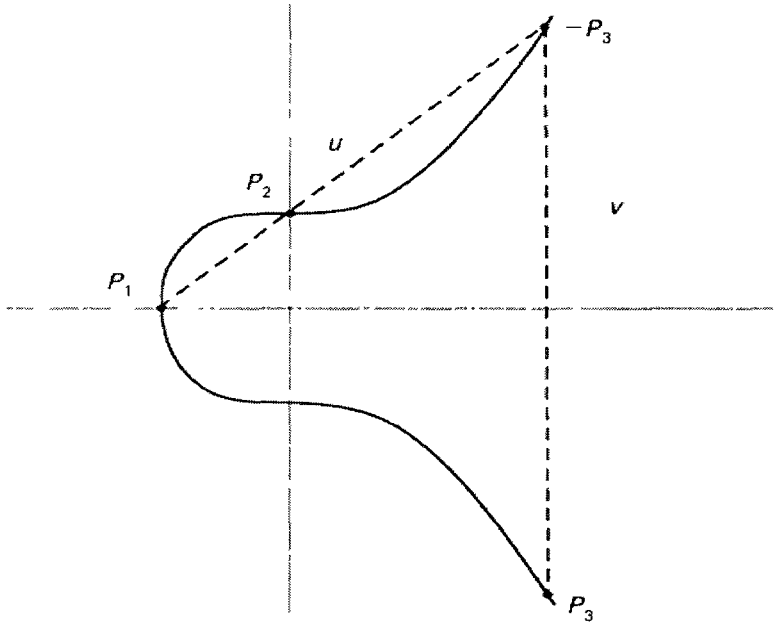

Figure4-3 an Illustration of u and v

From the definition we can see

$$div(u)=(P1)+(P2)+(-(P3))-3(O)$$

$$div(v)=(P3)+(-(P3))-2(O)$$

Note that multiplying rational functions corresponds to addition of their divisors and division of rational functions corresponds to subtraction of their divisors [33].

Therefore

$$div(u)-div(v)=div(u/v)=(P1)+(P2)-(P3)-(O)$$

Suppose we have two divisors:

$$D1=(P1)-(O)+div(f_1)$$

$$D2=(P2)-(O)+div(f_2)$$

where f1 and f2 are rational functions.

Then

$$D1+D2$$

$$=(P1)+(P2)-2(O)+div(f_1)+div(f_2)$$

$$=div(u/v)+(P3)+(O)-2(O)+dif(f_1f_2)$$

$$=(P3)-(O)+div(f_1f_2u/v) \quad (4.6)$$

The equation 4.6 provides a way to calculate the addition of two divisors on an ellpitc curve and keeps the result in the form of (P)-(O)+div(f). To compute the divisor l(P)-l(O) in the tate pairing, we can use the same method l times. However, in practice, l is a large number usually greater than 2^{112}, so the above method is not efficient. Miller provides an algorithm to compute l(P)-l(O) by a doubling and adding technique. The algorithm starts with f=1, S=P, and R is a random point on E with R≠O or P. After the first step, there is a doubling and adding iteration through the binary expansion of l. The doubling step is performed at each iteration and the adding step is performed if the bit of l is one. At each step, the (Q)-(O) is evaluated. Because the order of P is l, after l steps of doubling or adding the divisors, we can get a rational function f of the divisor l(P)-l(O). All the intermediate divisors involving points cancel each other after l steps.

To compute the divisor (Q)-(O) at which f is evaluated, we replace (Q)-(O) by the divisor (Q+R)-(R) where R is a random point on E, R≠O or P, Q+R≠P. The reason we can replace (Q)-(O) with (Q+R)-(R) is that these two divisors are equivalent. The proof is given below:

Suppose u is the line which passes through Q and R and v is the vertical line passing through (Q+R).

$$div(u)=(Q)+(R)+(-(Q+R))-3(O)$$

$$div(v)=(Q+R)+(-(Q+R))-2(O)$$

$$div(u)-div(v)=div(u/v)=(Q)-(O)+(R)-(Q+R)$$

$$(Q)-(O)=(Q+R)-(R)+div(u/v)$$

Since (Q)-(O) differs from (Q+R)-(R) by a rational function div(u/v), therefore they are equivalent.

4.2 Algorithms

In this section, we give a description of the algorithms used in our implementation. Our implementation borrowed ideas from the identity-based cryptography standard by Voltage Security [9] and some open source codes. We implement the elliptic curve $y^2=x^3+ax+b$ over Fp where p is a large prime (so the characteristic of the field Char(Fp)≠2,3). The major algorithms include the following parts: group computation on elliptic curves, the Tate Pairing calculation and pairing-based cryptography implementation. The algorithms for two-party key agreement implementation and extended policy based cryptography implementation can be found in the paring-based cryptography implementation.

4.2.1 Group Computation on Elliptic Curve

Algorithm 4.2.1.1 Point Doubling

Input:

- A point A on E(Fp) with A=(x,y)
- An elliptic curve E/Fp: $y^2=x^3+ax+b$ where a,b∈Zp and are set when the elliptic curve is generated

Output:

A point [2]A on E(Fp)

Method:

1. If E(Fp)==null or A==null, throw exception("parameters not set"), stop the program
2. If A==O or y==0, then return O
3. Let lambda = (3 * x^2+a) / (2 * y)
4. Let x' = lambda^2 - 2 * x
5. Let y' = (x - x') * lambda – y
6. Return (x', y').

Algorithm 4.2.1.2 Point Addition

Input:

- A point A on E(Fp) with A=(x_a,y_a)
- A point B on E(Fp) with B=(x_b,y_b)
- An elliptic curve E/Fp: $y^2=x^3+ax+b$ where a,b are parameters and can be set when the elliptic curve is generated

Output:

A point A+B on E(Fp)

Method:

1. If E==null or A== null or B== null, then throw exception("parameters not set") and stop the program
2. If A==O, return B
3. If B == O, return A
4. If x_a == x_b:
   a) If y_a == -y_b, return O
   b) Else if y_a==y_b, return [2]A computed using Algorithm 4.2.1.1 (PointDouble1)
5. Otherwise:
   a) Let lambda = (y_b - y_a) / (x_b - x_a)
   b) Let x' = lambda^2 - x_a - x_b
   c) Let y' = (x_a- x') * lambda - y_a
   d) Return (x', y')

Algorithm 4.2.1.3 Point Multiplication

Input:

- A point A on E(F_p)
- An integer l > 0
- An elliptic curve E/F_p: $y^2=x^3+ax+b$ where a,b are parameters and can be set when the elliptic curve is generated

Output:

The point [l]A on E(Fp)

Method:

1. If E==null or A==null, then throw exception("parameters not set"), stop the program
2. Let result=O
3. Let temp=O
4. For i=l.bitlength() downto 0 do
   a) temp=[2]result by algorithm 4.2.1.1
   b) test bit i, if l.testbit(i) == 1 then result=temp+ A by algorithm 4.2.1.2
   c) else result=temp
5. return result

4.2.2 Tate Pairing Calculation

Algorithm 4.2.2.1 (evalvertical_v): evaluate the divisor of a vertical line v on an elliptic curve E

Input:

- A point A(xa,ya) on an elliptic curve E
- A point B(xb,yb) on an elliptic curve E with B≠O
- An elliptic curve E/Fp: $y^2=x^3+ax+b$ where a,b∈Zp and are set when the elliptic curve is generated

Output:

- The divisor of the vertical line v going through A evaluated at B

Method:

1. If A== null or B== null or E==null, then throw exception("parameters not set") stop the program
2. If A == O, or B== O, then return 1
3. else return xb-xa

Algorithm 4.2.2.2 (evaltangent_u): evaluate the divisor of a tangent line u on an elliptic curve E

Input:

- A point A(xa,ya) on an elliptic curve E
- A point B(xb,yb) on an elliptic curve E with B≠O
- An elliptic curve E/Fp: $y^2=x^3+ax+b$ where a,b∈Zp and are set when the elliptic curve is generated

Output:

- The divisor of the tangent line u to A evaluated at B

Method:

1. If A ==null or E==null, then throw exception("parameters not set"), stop the program
2. If A==O, then return 1
3. If ya==0, then return the evaluation of vertical line going through A evaluated at B by using algorithm 4.2.2.1 evalvertical_v(A,B)
4. else let $m=(3*xa^2+a)/2*ya$
5. return yb-ya-m*(xb-xa)

Algorithm 4.2.2.3 (eval_u): evaluate the divisor of a line u on an elliptic curve E

Input:

- Two points A(xa,ya), A'(xa',ya') on an elliptic curve E

- A point B(xb,yb) on an elliptic curve E with B≠O
- An elliptic curve E/Fp: $y^2=x^3+ax+b$ where a,b∈Zp and are set when the elliptic curve is generated

Output:

- The divisor of the line u going through A and A' evaluated at B

Method:

1. If A== null or A' == null or B== null or E== null, then throw exception("parameters not set"), stop the program
2. If A== O, then return evalvertical_v(A',B) using algorithm 4.2.2.1
3. If A' == O or A+A' == O , then return evalvertical_v(A, B) using algorithm 4.2.2.1
4. If A== A', then return evaltangent_u(A, B) using algorithm 4.2.2.2
5. else let m=(ya'-ya)/(xa'-xa), return yb-ya-m(xb-xa)

Algorithm 4.2.2.4 (Miller's): Calculating the Tate pairing using Miller's Algorithm

Input:

- A point P with order n on an elliptic curve E and n divides #E(Fp)
- A point Q with order n on an elliptic curve E and n divides #E(Fp)
- An elliptic curve E/Fp: $y^2=x^3+ax+b$ where a,b∈Zp and are set when the elliptic curve is generated

Output:

- The Tate pairing e(P,Q)

Method:

1. If A ==null or B==null or E==null, then throw exception("parameters not set"), stop the program

2. Let f=1, t=n.bitlength()-1, S=P
3. Choose a random point R on E, R≠O or P, Q+R≠O or P
4. For i=t downto 0
    1) Use algorithm 4.2.2.2 to calculate tangent line at the point of S, evaluate u at the point of Q+R and evaluate u at the point of R. Denote us,s(Q+R)= evaltangent_u(S,Q+R) and us,s(R)= evaltangent_u(S,R).
    2) Use algorithm 4.2.2.1 to calculate line v for doubling S, evaluate v at the point of Q+R and evaluate v at the point of R. Let v2s(Q+R)=evalvertical_v (2S,Q+R) and v2s(R)= evalvertical_v (2S,R)
    3) f= f^2 (us,s(Q+R)v2s(R)/v2s(Q+R)us,s(R))
    4) if n.testbit(i) ==1
        a) use algorithm 4.2.2.3 to calculate line u going through S and P, evaluate u at the point of Q+R and evaluate u at the point of R. Let us,p(Q+R)=eval_u(S,P,Q+R) and us,p(R)=eval_u(S,P,R).
        b) use algorithm 4.2.2.1 to calculate line v for the addition of S and P, evaluate v at the point of Q+R and evaluate v at the point of R. Let vs+p(Q+R) = evalvertical_v(S+P,Q+R) and vs+p(R) = evalvertical_v(S+P,R).
        c) f=f(us,p(Q+R)vs+p(R)/vs+p(Q+R)us,p(R))
        d) S=S+P
5. return f.

4.2.3 Pairing Based Cryptography Algorithm

Algorithm 4.2.3.1 (Maptopoint): Map a string to a point on an elliptic curve

Input:

- a string m
- a cryptographic hash function H
- An elliptic curve E/Fp: $y^2=x^3+ax+b$ where $a,b \in Zp$ and are set when the elliptic curve is generated

Output:

- A point in E/Fp

Method:

1. If m== null or H== null E== null, then throw exception("parameters not set"), stop the program
2. Use H to generate a message digest md for m, md=H(m)
3. Generate a biginteger value from md using java.math.BigInteger
4. value=value modulo p
5. If a==0 then
   a) y=value
   b) $x=(y^2-b)^{(2p-1)/3}$ modulo p
6. else
   a) x=value
   b) alpha=(x3+ax+b) modulo p
   c) if alpha== 0, then return (x,0)

d) else

d.1) lengre=(alpha)$^{(p-1)/2}$ modulo p

d.2) if lengre 1 then y=(alpha)$^{1/2}$ mod p

d.3) else return null

7. return (x,y)

Algorithm 4.2.3.2 (System parameters setup): Initialize the parameters of the cryptographic scheme

Input:

- A large prime p
- A large prime q where q is the order of the subgroup of E/Fp
- A cryptographic hash function H
- An integer a
- An integer b

Output:

- System Parameters {E/Fp,p,q,Gs,s} where G is the order of the subgroup of E/Fq, Ps is the public key of the TA and s is the master key of The TA

Method:

1. If one of the parameters of the input is null, throw exception("parameters not set"), stop the program
2. If $(-16(4a^3+27b^2))==0$ then throw exception("a,b not set correctly"), stop the program.
3. Let E: $y^2=x^3+ax+b$
4. select a random point G'(x,y) $\in$ E/Fp

5. Use 4.2.1.3 Point Multiplication to compute [q]G'
6. If [q]G' == O, then G=G'
7. else go back to 4
8. Generate a random number s between[2,p-1]
9. Let Ps=[s]G using algorithm 4.2.1.3 Point Multiplication
10. return (E/Fp,p,q,G,s,Ps)

Algorithm 4.2.3.3.(Two party key agreement protocol): Generate a shared key between two communicating parties

Input:

- An identity string ida
- An identity string idb

Output:

- A shared secret Kab where Kab=e(privatekey(ida), publickey(idb))

Method:

1. If one of the parameters : {ida,idb} is null, throw exceptions("parameters not set"), stop the program
2. Use algorithm 4.2.3.2 to get system parameters E/Fp, p,q and s
3. Use algorithm 4.2.3.1 to map ida to a point publickey(ida) on E
4. If publickey(ida) == null, then throw exception("cannot map points"), stop the program
5. Use algorithm 4.2.3.1 to map idb to a point publickey(idb) on E
6. if publickey(idb) == null, then throw exception("cannot map points"), stop the program

7. Generate a point privatekey(ida) on E with privatekey(ida)=[s]publickey(ida) by using algorithm 4.2.1.3 Point Multiplication
8. Compute Kab=e(privatekey(ida), publickey(idb)) by using algorithm 4.2.2.4 Miller's algorithm to calculate the tate pairing
9. Return Kab

Algorithm 4.2.3.4 (Extended PBC: public key generation): Generate the public key for the end user

Input:

- A random number sa chosen by the end user as his secret

Output:

- Public key pair <Xpol, Ypol>

Method:

1. If sa ==null, then throw exception("parameters not set"), stop the program
2. else get the system parameters {E/Fp,p,q,G,s,Ps} by using algorithm 4.2.3.2
3. Compute Xpol=[sa]G on E by using algorithm 4.2.1.3 Point Multiplication
4. Compute Ypol=[sa] Ps on E by using algorithm 4.2.1.3 Point Multiplication
5. Get the Tate pairing e(Xpol, Ps) by using algorithm 4.2.2.4 Miller's algorithm
6. Get the Tate pairing e(Ypol, G) by using algorithm 4.2.2.4 Miller's algorithm
7. if e(Xpol, Ps) == e(Ypol,G) , return (Xpol, Ypol)
8. else return null

Algorithm 4.2.3.4 (Extended PBC:Mappolicy): Map a policy to a point on an elliptic curve

Input:

- A list L with each item of L representing a condition of the policy for the end user

Output:

- A point on E corresponding to the policy

Method:

1. If L== null or L.length== 0, throw exception("parameters not set"), stop the program
2. get the system parameters {E/Fp,p,q,G,s,Ps} by using algorithm 4.2.3.2
3. result== O
4. For i=0 to L.length-1
   a) Use algorithm 4.2.3.1 Maptopoint to map each condition L[i] to a point L' on E
   b) Use algorithm 4.2.1.2 Point Addition to compute result=result+L[i]
5. Return result

Algorithm 4.2.3.5 (Extended PBC: Private key generation): generate the private key for the end user according to his access policy and his own secret

Input:

- A list L with each item of L representing a condition of the policy for the end user
- A secret sa

Output:

- The end user's private key Psa which is a point on E

Method:

1. If L== null or sa== 0, throw exception("parameters not set"), stop the program
2. get the system parameters {E/Fp,p,q,G,s,Ps} by using algorithm 4.2.3.2
3. Use algorithm 4.2.3.4 to get the point L' on E corresponding to the policy
4. If L' == O, then throw exception, stop the program
5. Use algorithm 4.2.1.3 Point Multiplication to compute Psa=[sa][s]L'

6. return Psa

Algorithm 4.2.3.6 (Extended PBC: Encryption): Encrypt a message according to the end user' policy and corresponding public key

Input:

- The end user's access policy L
- The corresponding public key (Xpol, Ypol)
- Message M in bytes M[]

Output:

- A cipertext (U,V)

Method:

1. If one of the inputs is null, throw exception("parameters not set"), stop the program
2. get the system parameters {E/Fp,p,q,G,s,Ps,H} by using algorithm 4.2.3.2
3. Use algorithm 4.2.3.4 to map the policy L to a point L' on E/Fp
4. If L' == O, then throw exception, stop the program
5. Get the Tate pairing T=e(L', Ypol) by using algorithm 4.2.2.4 Miller's algorithm
6. If T ==1, then throw exception, stop the program
7. Choose a random number $r \in Z_p$
8. If rG ==O, then go back to 7, else U=rG
9. $T=T^r$ modulo p
10. Get the hashcode of T and Convert the result to a large number
11. Convert the result from 8 to bytes and set the result to T'
12. Let V be a bytearray and V.length=M.length
13. Use algorithm 4.2.3.8 BlockXor to compute V=BlockXor(M, T')

14. return (U,V)

Algorithm 4.2.3.7 (Extended PBC: Decryption): Decrypt a message using the end user's private key

Input:

- The corresponding private key Psa
- Cipertext (U,V)

Output:

- Message M

Method:

1. If one of the inputs is null, throw exception("parameters not set"), stop the program
2. Get the system parameters {E/Fp,p,q,G,s,Ps,H} by using algorithm 4.2.3.2
3. Get the Tate pair e(Psa,U) using the algorithm 4.2.2.4 Miller's algorithm
4. Get the hashcode of e(Psa,U) and convert the result to a big number
5. Get the byte result of 4 and set the result to a bytearray W
6. Use algorithm 4.2.3.8 to compute M=BlockXor(V,W)
7. Return M

Algorithm 4.2.3.8 (BlockXor): Xor two bytearrays and return the result in a bytearray

Input:

- A bytearray A
- A bytearray B

Output

- A bytearray C=A Xor B

Method:

1. If any of the inputs is null, throw exception("parameters not set"), stop the program
2. Let blocksize=B.length
3. Let C.length=A.length
4. For i=0; i<C.length; i+=blocksize

   For j=0; j<B.length && (i+j)<C.length; j++

   C[i+j]=A[i+j] xor B[j]
5. Return C

4.3 Test Results

In this section, we give the test results of our implementation. The test vectors we use are from [23], [9] and [33]. The test environment is listed below:

- Hardware: Sony Laptop VGN-FE Series: 1.66GHz/768 Memory/80G Hard Drive
- Operation System: Window xp with service pack 2
- Software Platform: Eclipse PlatformVersion: 3.4.1
- Programming language: Java

4.3.1 Group Computation on Elliptic Curve Test Result

Test parameters:

- E/F_{29}: $y^2=x^3+4x+20$
- Point under test $A=(1,5) \in E/F_{29}$

Expected Result:

- $2A=(4,19)$
- $5A=(6,12)$

- 7A=(24,22)
- 37A=O

Test Result:

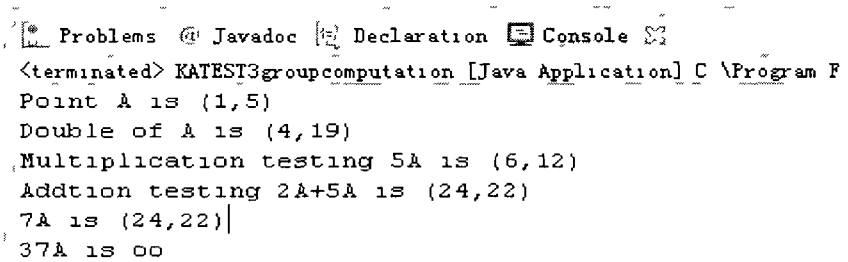

Figure4-4 Group Computation on E/F_{29} Test Result

4.3.2 Tate Pairing Calculation Test Result

Test parameters:

- E/F_{11}: $y^2=x^3+x$
- Point tested P=(5,3) E/F11[3]
- Point under test Q=(7,8)
- Random Point R=(10,3)

Expected Result:

- e(P,Q)=7

Test Result:

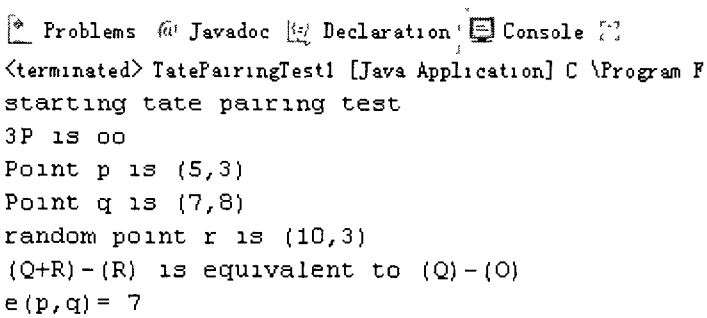

Figure4-5 Tate Pairing Calculation Test Result

4.3.3 Two-Party Key Agreement Protocol Test Result

Test Group1: Small Prime Number Test

Test parameters:

- E/F_{43}: $y^2=x^3+x$
- The order of the subgroup of E/F_{43} is 11
- Ida="agent1@container1"
- Idb="agent2@container1"
- Hash Function: MD5
- Master key :7

Expected Result:

- $Kab=Kba \in F_{43}^2$

Test Result:

```
Problems  @ Javadoc  Declaration  Console
<terminated> KATEST [Java Application] C \Program Files\Java\j
testing ....
Agent1's private key is(13,24)
Agent2's public key is(14,36)
shared key Kab=1731
Agent2's private key is(4,38)
Agent1's Public key is (31,18)
shared key Kba=1731
```

Figure4-6 Small Number Two-Party Key Agreement Protocol Test Result

Test Group2: Large Prime Number Test

Test parameters:

- E/F_p: $y^2=x^3+x$

- p=0xbfffffffffffffffffffffffffcffff3
- q=0xffffffffffffffffffffffffffbffff, q is the order of the subgroup of E/Fp
- Ida="agent1@container1"
- Idb="agent2@container1"
- Hash Function: MD5
- Master key: a random number between [0,p-1]
- A random generator G ∈ E/Fp

Expected Result:

- Kab=Kba $\in F_p^2$
- Time to compute Kab in milliseconds
- Time to compute Kba in milliseconds

Test Result:

```
Problems  Javadoc  Declaration  Console
<terminated> katest2 [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (2010-6-26 下午12:45:29)
testing ....
master key is 107619145813164463899379732426487308257l
The generator G is (247995389159869722883213126869290106168l,2188406533197355312535317747328775250905)
Agent1's private key is (185057262451737377962913775880482470604 9,387519625972785597978633593969341394880 4)
Agent2's public key is (283141207867494820277725566102330528895 3,65702120604177558876952810932633632210 9)
shared key Kab= 98286466983448183953105706098963452082340014707492028279476294528644689310446 80
Time to compute Kab is 657ms
Agent2's private key is (32602220091643731803063790157429193830 39,20458381320776921103857593104281435319 94)
Agent1's public key is (34268145636866520191190279224140347302 81,13604539820587939447947230707854422006 9)
shared key Kba= 98286466983448183953105706098963452082340014707492028279476294528644689310446 80
Time to compute Kba is 625ms
```

Figure4-7 Large Prime Two-Party Key Agreement Protocol Test Result

4.3.4 Extended PBC Test Result

Test Group1: Small Prime Number Test

Test Parameters:

- E/F_{43} : y2=x3+x
- The order of the subgroup of E/F_{43} is 11
- TA's master key s=7
- The end user's secret sa=2
- The group generator G=(23,8)
- The access policy L=condition C1 and condition C2. C1 and C2 can represent the access conditions according to a specific situation. For example C1 can be "Doctors in Group1" and C2 can be "Patients in Group2". Thus the access policy L is "Doctors in group1 can access the medical records of patients in Group2".
- The Message encrypted "This is a test."

Test1: Public Key Generation

Expected Result1:

- Output public key <Xpol,Ypol> where Xpol=[sa]P and Ypol=[s]Ps
- $e(Xpol,Ps)=e(Ypol,P) \in F_{43}^2$

Test Result1:

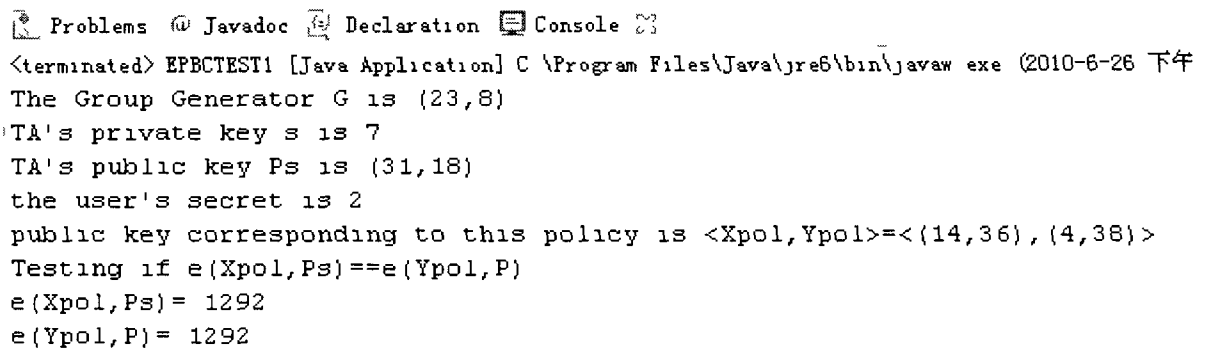

Figure4-8 Small Prime Extended PBC Public Key Generation Test Result

Test2: Map Policy

Expected Result2:

- A point on E/F_{43} representing the policy L

Test Result2:

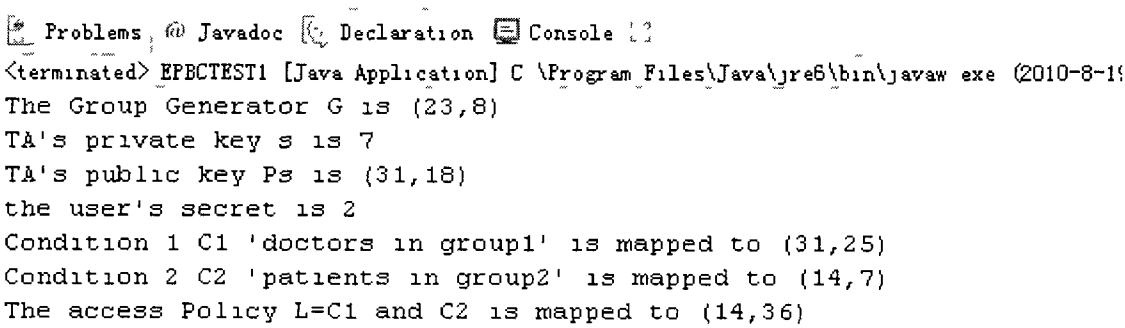

Figure4-9 Small Prime Extended PBC Map Policy Test Result

Test3: Private Key Generation

Expected Result3:

- A point Psa on E/F_{43} where Psa=[sa*s]L' where L' is the point on E corresponding to the access policy

Test Result3:

```
Problems  Javadoc  Declaration  Console
<terminated> EPBCTEST1 [Java Application] C \Program Files\Java\jre6\bin\ja
The Group Generator G is (23,8)
TA's private key s is 7
TA's public key Ps is (31,18)
the user's secret is 2
Condition 1 C1 'doctors in group1' is mapped to (31,25)
Condition 2 C2 'patients in group2' is mapped to (14,7)
The access Policy L=C1 and C2 is mapped to (14,36)
The private key Psa (13,19)
```

Figure4-10 Small Prime Extended PBC Private Key Generation Test Result

Test4: Encryption/Decryption

Expected Result4:

- H2(e(L', Ypol)^r)= H2(e(Xpol,Psa)) where L' is the mapped point of the access policy
- Cipertext V
- Decrypted message " This is a test"

Test Result4:

```
Problems  Javadoc  Declaration  Console
<terminated> EPBCTEST1 [Java Application] C \Program Files\Java\jre6\bin\javaw exe
The Group Generator G is (23,8)
TA's private key s is 7
TA's public key Ps is (31,18)
the user's secret is 2
random number r is 3
U=[r]P is (4,38)
e(L',Ypol)^r = 362
e(Psa,U)= 362
Ciphertext is u┐h├╵└rJ`Ju%r
Decrypted String is this is a test
```

Figure4-11 Small Prime Extended PBC Encryption/Decryption Test Result

Test Group2: Large Prime Number Test

Test Parameters:

- $E/F_p : y^2=x^3+x$
- p=0xbfffffffffffffffffffffffffcffff3
- q=0xfffffffffffffffffffffffffbffff, q is the order of the subgroup of E/Fp
- Hash Function: MD5
- Master key: a random number between [0,p-1]
- User secret: a random number between [0,p-1]
- A random generator $G \in E/Fp$
- The access policy L=condition C1 and condition C2. C1 and C2 can represent the access conditions according to a specific situation.
- The Message encrypted "This is a test."

Test1: Public Key Generation

Expected Result1:

- Output public key <Xpol,Ypol> where Xpol=[sa]P and Ypol=[s]Ps
- $e(Xpol,Ps)=e(Ypol,P) \in F_p^2$

Test Result1:

```
Problems  Javadoc  Declaration  Console
<terminated> EPBCTEST2 [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (2010-6-27 上午02:31:30)
The Group Generator G is (327250259024023337431309529602287829652 1,979537829093985999342080202689721400314)
TA's private key s is 546195174443504532288361388849624785393
TA's public key Ps is (276428037225510431601172468929369716880 9,156989938438223348608427554360666854592 0)
the user's secret is 784523114042565069021012712004376199273
public key corresponding to this policy is
Xpol=
(900941634843408713136336542038472502249,270381849399781665502579525056219333802 1)
Ypol=
(362645462996088557451436298850416801137,129791851465607735170927692062843408426 2)
Testing if e(Xpol,Ps)==e(Ypol,P)
e(Xpol,Ps)= 303439340816994165047772560807404069271364825767331833460412571557341490998297 2
e(Ypol,P)= 303439340816994165047772560807404069271364825767331833460412571557341490998297 2
```

Figure4-12 Large Prime Extended PBC Public Key Generation Test Result

Test2: Map Policy

Expected Result2:

- A point on E/F_p representing the policy L

Test Result2:

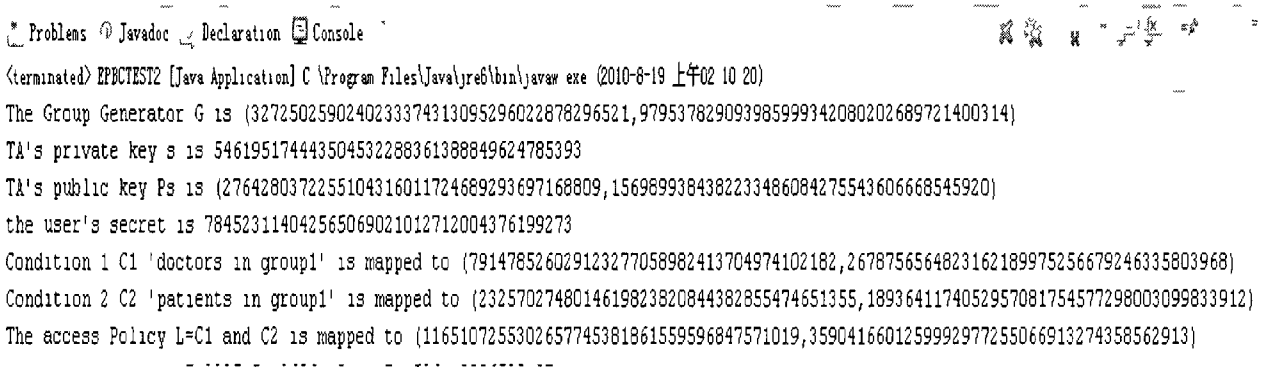

Figure4-13 Large Prime Extended PBC Map Policy Test Result

Test3: Private Key Generation

Expected Result3:

- A point Psa on E/F_p where Psa=[sa*s]L' where L' is the point on E corresponding to the access policy

Test Result3:

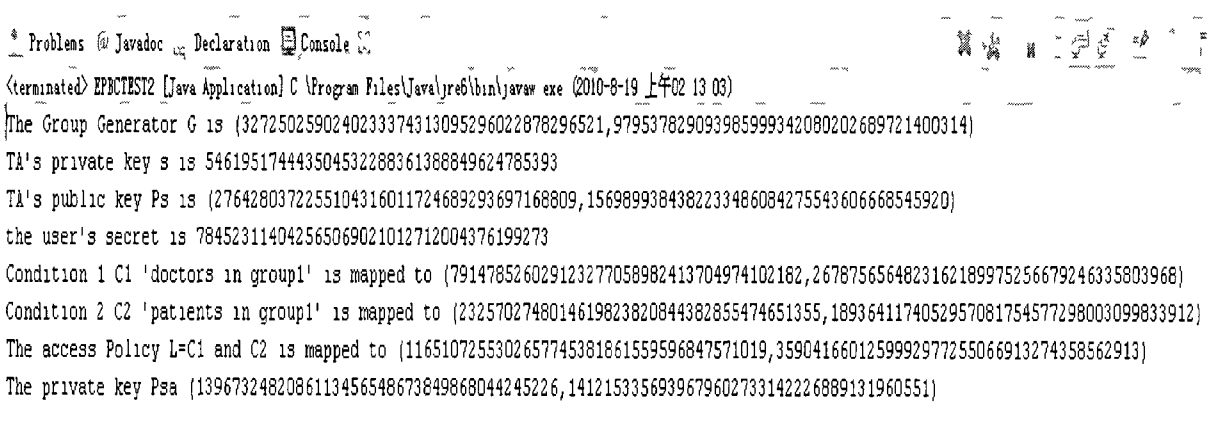

Figure4-14 Large Prime Extended PBC Private Key Generation Test Result

Test4: Encryption/Decryption

Expected Result3:

- H2(e(L', Ypol)^r)= H2(e(Xpol,Psa)) where L' is the mapped point of the access policy
- Cipertext V
- Decrypted message " This is a test"
- Encryption time in milliseconds
- Decryption time in milliseconds

Test Result4:

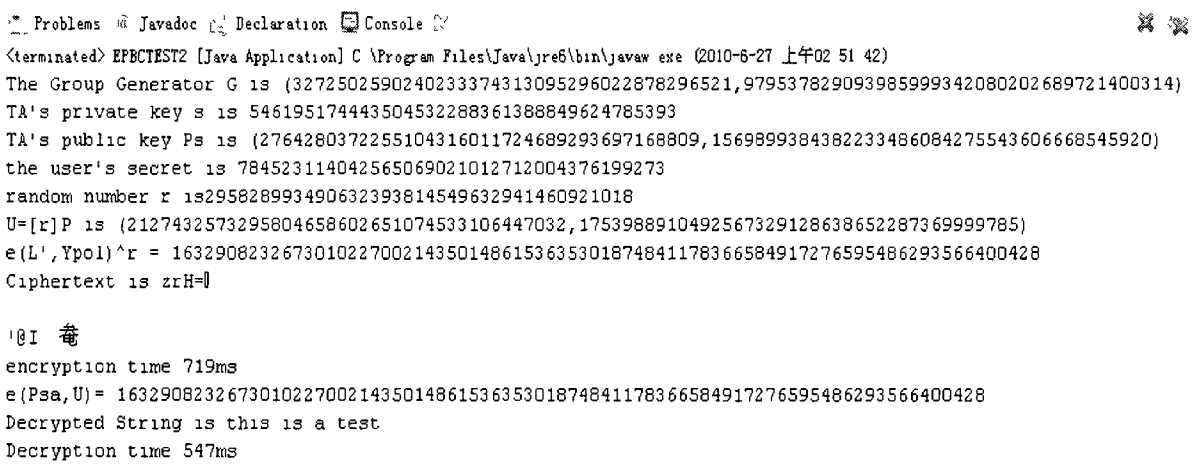

Figure4-15 Large Prime Extended PBC Encryption/Decryption Test Result

Test Group3: Large Prime Number Long Policy Test

In this test group, we test the encryption and decryption time under the policy L which is composed of multiple conditions. The test parameters in this group test are the same as those in test group 2 except that in this test group, the policy L is composed of multiple conditions. We want to show that the encryption and decryption time under L shouldn't vary too much

compared with the encryption/decryption time under a policy which is composed of a small number of conditions.

Test Parameters:

- Policy $L1=\Sigma_{i=1}^{2} C_i$ where C_i is the condition
- Policy $L2=\Sigma_{i=1}^{10} C_i$ where C_i is the condition
- Policy $L3=\Sigma_{i=1}^{20} C_i$ where C_i is the condition
- Other parameters are the same as in Test Group 2

Expected Results:

- Key Pair generation time in milliseconds under L1, L2 and L3
- Encryption/Decryption time in milliseconds under L1,L2 and L3
- Encryption/Decryption time under the three policies shouldn't vary too much. In our test case they all should be less than or around 1s.

Test Results:

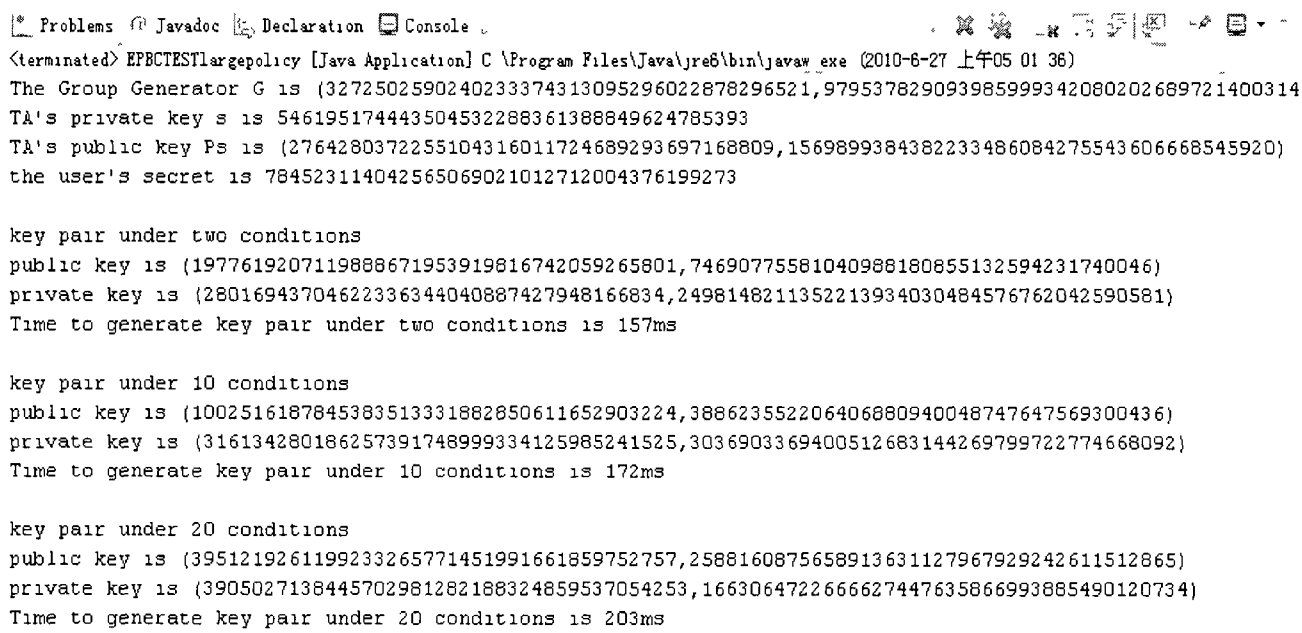

Figure4-16 Large Prime Number Long Policy Key Pair Generation Time Test Result

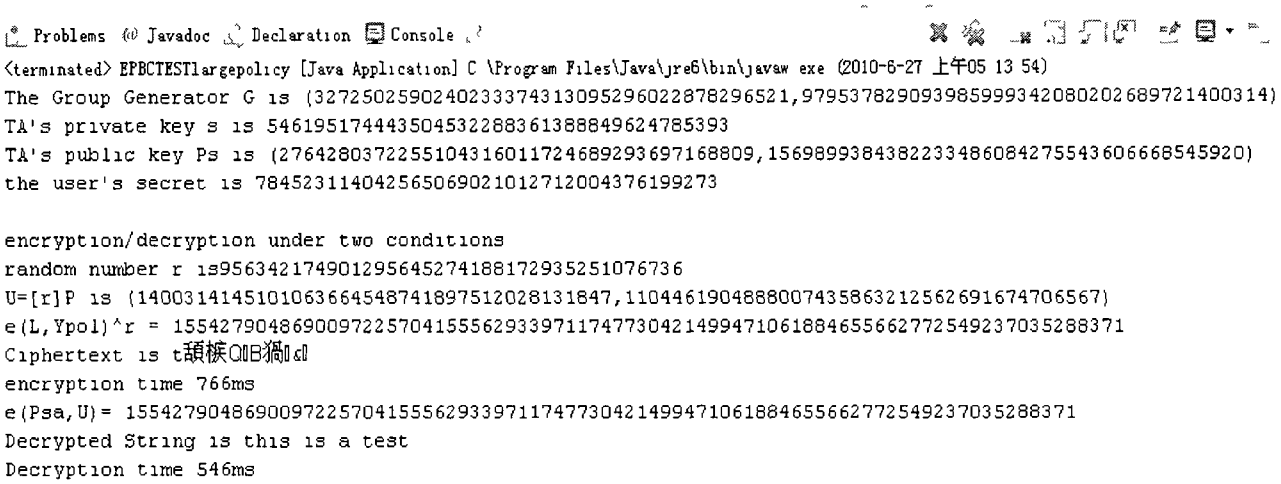

Figure4-17 Large Prime Number Encryption/Decryption Time Under Two conditions Test Result

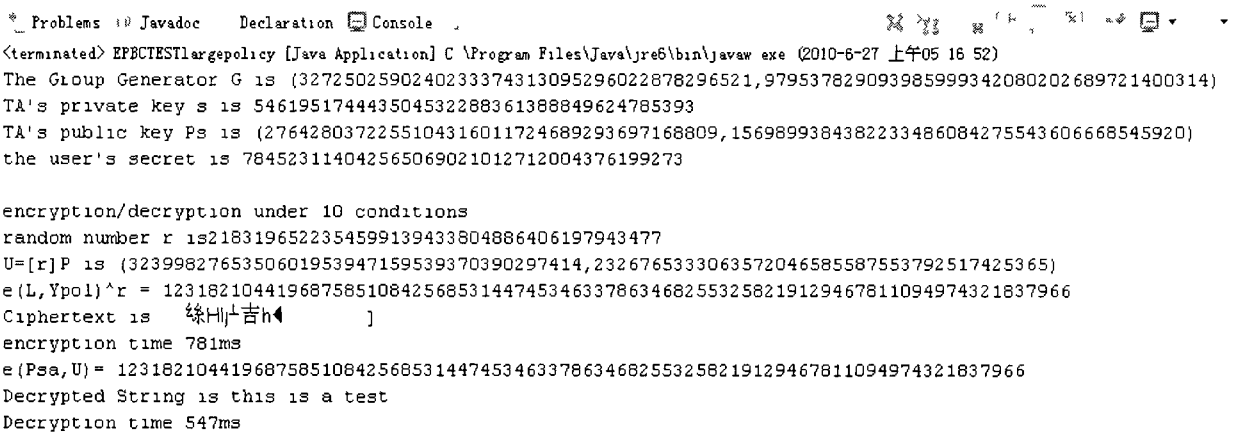

Figure4-18 Large Prime Number Encryption/Decryption Time Under 10 conditions Test Result

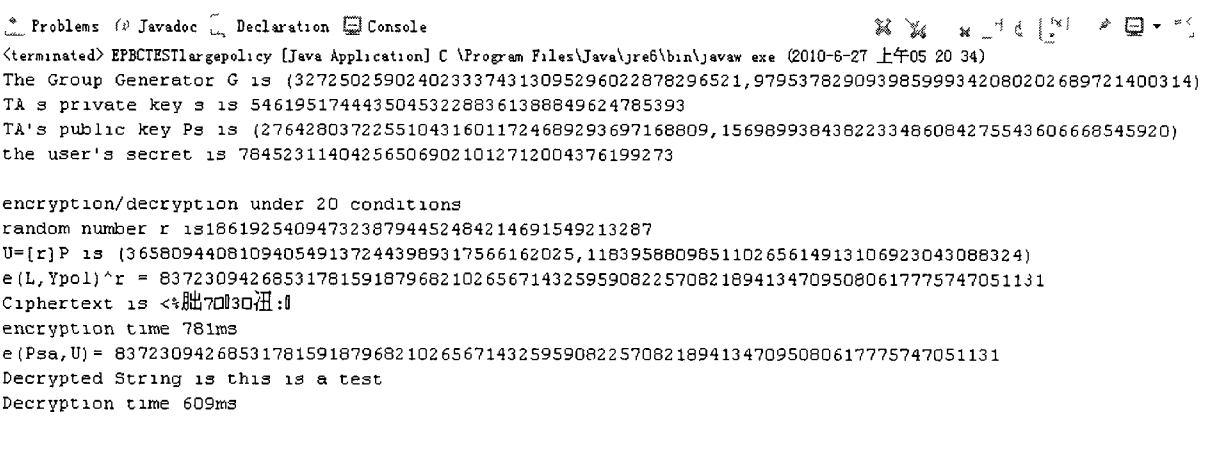

Figure4-19 Large Prime Number Encryption/Decryption Time Under 20 conditions Test Result

4.4 Summary

This chapter describes the implementation of paring-based cryptography, namely the two-party key agreement protocol and our extended policy-based encryption scheme. We use the Java technologies to accomplish the implementation. The chapter first describes the mathematical background of elliptic curves and the Tate pairing. It then discusses the algorithms used in our implementation. The algorithms include three major parts: group computation on elliptic curves, the Tate pairing calculation and the pairing-based cryptography implementation. The test results are given following the algorithms. The test vectors used to test the program are divided into two groups: a small prime number test and a large prime number test. For our extended policy-based encryption scheme, we also test encryption and decryption under longer policies. Our test results show that the encryption and decryption times don't vary too much under different policies which are composed of different numbers of conditions. In the test of our extended policy based cryptographic scheme, we use a string as a message to test the encryption and decryption. However, in reality, the message should be a random number which is a session key and is used in the later stage of real message transmission.

Chapter 5 Conclusions and Future Work

5.1 Summary and Conclusions

This thesis discusses the design and implementation of a security architecture for a mobile agent based e-hospital system. The whole architecture is a general one and our cryptographic schemes, namely two party key agreement protocol or the extended policy based cryptographic scheme, can be used specifically to address the security problems in mobile agent system. Two party key agreement protocol or our extended policy based cryptographic scheme uses the fact that only legal parties can have enough credentials to compute securely. The cryptographic schemes used in our design can ensure secure computation in mobile agent systems whether agents move from platforms to platforms or not.

We approach the security problems in this e-hospital from the risk analysis of the agent-based system, the functional framework for the e-hospital, the requirements from the related regulations and the security policy from the hospital. The whole approach is guided by ISO17799, an international standard for information security management systems. The whole e-hospital system is divided into three domains according to their security levels. Security protection mechanisms have been proposed for each domain.

Two cryptographic schemes have been proposed to address the network security and end users' access control. For end users' access control, we extend the policy-based cryptographic scheme and address the non-repudiation problem in the original policy-based cryptographic scheme. The security analysis is given following the design to show how the design meets the security objectives for this e-hospital. The full scheme of our extended

policy-based cryptography is also analyzed to give an intuition of why it is adaptive chosen cipher attack secure.

As mentioned before, ISO17799 is a general standard for information security management systems. It is not a specific standard for mobile agent systems. The reason we use ISO17799 as our guide is that this e-hospital is still inside of the hospital network and parts of the components, such as the database server, of this e-hospital are still inside of the centralized system. Therefore we need to consider the protection for the centralized system and the mobile agent system as well. For the protection of the mobile agent system, our two cryptographic schemes work very well to address the security problems inside of mobile agent systems. As we mentioned at the beginning of the thesis, there are four types of possible attacks in mobile agent systems and secure computation is the security aim for a mobile agent system. Two party key agreement protocol addresses the secure computation based on the fact that only two authorized parties (agents or platforms) can compute the shared secret key and thus can communicate securely.

In the design of the security architecture, we address the end user access control by extended policy based cryptography. We do not follow the traditional access control in centralized systems in which a server decides end users' access control. The reasons are listed below:

- Access policies for end users in this e-hospital are complicated and it is hard to set access control rules in the database or network devices;
- In the traditional access control, a centralized reference monitor should be accessed for authorization purposes. In the scenario of this e-hospital, since it is a mobile agent system, an agent moves from platform to platform to finish the job on behalf of end users.

If we use traditional access control mechanism in this e-hospital, agents need to be authenticated and authorized every time they move and the authorization server could be the bottleneck of the whole system. If the authorization server cannot be accessed because of the network problems or because the server is down, the whole authorization fails;

- Extended policy based cryptography provides a better way to achieve access control in distributed systems especially in mobile agent systems when agents move. As long as the end user possesses the corresponding credentials, the agent, on behalf of the end user to finish the task, can decrypt. In this way, the end user access control can be ensured.
- Extended policy based cryptography also protects end users' privacy better than the traditional access control mechanism. The basic idea behind the extended policy based cryptography is that as long as the end user complies with the policy, he/she can access the data. The data owner does not care about the private information of the end user. In traditional access control mechanism, the end user needs to reveal some private information to the system such as his name or his account to access the data even though the access control rules may not need this information.

For the implementation, we initially considered using public libraries for the elliptic curve operations. However, we found it difficult to map an identity to a point on an elliptic curve using the public library for the Tate pairing. Consequently, we referred to the implementation standard listed in [9] and wrote the code ourselves.

We implement the cryptographic parts of our design using Java. Our implementation is based on elliptic curves defined over large prime. We provide the algorithms used in our

implementation, namely group computations on elliptic curves, the Tate pairing calculation and the pairing-based cryptographic implementation.

We use a small prime and a large prime to test the functionality and performance of the algorithms. The test results show that the encryption and decryption times do not vary too much under different lengths of access policies. The test results also show under the large prime, the computing time of two-party key agreement is less than one second. The encryption and decryption time under a large prime and different length of policies is also less than one second and should be acceptable for most hospital environments.

5.2 Future Work

As stated before, the end devices accessing the e-hospital can be laptops, PDAs and mobile phones. We only implement the cryptographic scheme for this e-hospital on laptops. The current version of our implementation is based on elliptic curves defined over a large prime in affine coordinates and can be improved to use other coordinates in the future. Realizing our work is just a starting point for this project, we outline the future work as follows.

- We only implement the cryptographic scheme on laptops; in the future we need to consider implementing the scheme on other platforms such as PDAs and mobile phones. Java 2 Platform, Micro Edition(J2ME) provides a Java platform for embedded devices such as mobile phones. In the future, we need to implement our cryptographic scheme using J2ME technologies.
- In the current version of our implementation, the large prime used is from the test vectors in [9]. In order to achieve different levels of security, we need to extend our implementation to generate large primes of different bit lengths. In [9] , they give the

bit length of parameters of n,p and q where n is security parameter , p is the order of E/Fp and q is the order of its subgroup E/Fq. The parameter of n must be one of 1024, 2048, 3072, 7680 or 15360, which corresponds to the equivalent bit security levels of 80, 112, 128, 192 and 256 bits respectively. The bit length of n,p and q is given as follows.

- If n = 1024, then let p = 512, q = 160
- If n = 2048, then let p = 1024, q = 224
- If n = 3072, then let p = 1536, q = 256
- If n = 7680, then let p = 3840, q = 384
- If n = 15360, then let p = 7680, q = 512

- The performance of elliptic curve cryptography is always a big concern. In [43], they show that elliptic curves defined over characteristic two and three can get good performance when the elliptic curve exponentiation is implemented directly in hardware. In the future, we can extend our implementation to elliptic curves over characteristic two and three.
- In [23], they give algorithms for group computation of elliptic curves under different coordinate systems to increase the performance. In the future, we can implement elliptic curves in other coordinate systems such as the projective coordinate system.
- In current version of our implementation, the policy is just a combination of strings which represent the conditions. In the future, we need to find an efficient way to represent policies.
- The e-hospital is a decision making system. Data mining technologies are used in this e-hospital. We use cryptography to protect the original data in the database but the

data mining tools still can access the private information. In the future, we need to consider techniques to de-identify the private information in the database before giving it to the data mining tools.

Bibliography

[1] S.S. Al-Riyami and K.G.Paterson. "Certificateless public key cryptography". In advances in Cryptography-ASIACRYPT 2003. 452-473 Springer-Verlag, LNCS 2894, 2003

[2] G. Appenzeller and B.Lynn. "Minimal-overhead IP security using identiy-based encryption". Technical Report, Voltage Inc. 2002

[3] W.A. Arbaugh, J.Katz and A.Khalili. "Toward secure key distribution in truly ad-hoc networks". In proc. 2003 Symposium on Applications and the Internet Workshop. I EEE Computer Society, 2003

[4] W.Bagga and R.Molva. "Policy-Based Cryptography and Applications". In Proceedings of Financial Cryptography and Data Security (FC'05), volume 3570 of LNCS, pp. 72-87, Springer Berlin/Heidelberg, 2005

[5] F. Bellifemine, G.Caire et al. "JADE: A software framework for developing multi-agent applications". In Information and Software Technology, Volume 50, Issues 1-2, Pages 10-21, 2008

[6] D. Boneh, B.Lynn and H.Shacham. "Short signatures from the Weil pairing". In advances in Cryptology-ASIACRYPT 2001. LNCS 2248,2001, 514-532. Springer-Verlag.

[7] D. Boneh and M.Franklin. "Identity-Based Encryption from the Weil Pairing". In Proceedings of Crypto 2001, volume 2139 of LNCS, pages 213-229. Springer, 2001.

[8] D. Boneh, W.Du and N.Li. "Oblivious signature-based envelope." In Proceedings of the 22nd annual symposium on Principles of distributed computing, pages 182–189. ACM Press, 2003.

[9] X Boyen and L Martin "Identity-based Cryptography standard (IBCS) Voltage Security, December, 2007

[10] L Chen, K Harrison, A Moss, N P Smart and D Soldera "Certification of Public Keys within an Identity Based System" Proceedings of the 5th International Conference on Information Security, pp 322–333 September 2002

[11] L Chen, C Kudla "Identity Based Authenticated Key Agreement Protocols from Pairings" Trusted systems Laboratory HP Laboratories Bristol HPL-2003-25 February 12th 2003

[12] D Chess, C Harrison and A Kershenbaum "Mobile Agents Are They a Good Idea?" IBM Research report, J Watson Research Center, March 1995

[13] G Durfee and D K Smetters "Domain-based administration of identity-based cryptosystems for secure email and IPSEC" In Proc 12th USENIX Security Symposium, 215-229, 2003

[14] R Fraser "ISO 27799 Security management in health using ISO/IEC 17799" CIHI Partnership Conference, June 6, 2006

[15] G Frey, M Muller and H -G Ruck "The Tate pairing and the discrete logarithm applied to elliptic curve cryptosystems" IEEE Trans Inf Theory, 45, 1717-1719, 1999

[16] S D Galbraith, K Harrison and D Soldera "Implementing the Tate pairing" , Trusted E-Services Laboratory HP Laboratories Bristol, HPL-2002-23, March 8th , 2002

[17] S D Galbraith "Supersingular curves in cryptography" In C Boyd(ed) ASIACRYPT 2001,Springer LNCS 2248 (2001) 495-513

[18] C. Gentry, J.Kempt and A.Silverberg. "Securing IPv6 neighbor discovery using address based keys". Internet Draft Document expired December 2002. Available from http://www.docomolabs-usa.com/pdf/PS2003-080.pdf

[19] C.Gentry. "Certificate-based encryption and certificate revocation problem". In advances in Cryptology-EUROCRYPT 2003. 272-293 Springer-Verlag, LNCS 2656, 2003.

[20] V. Goyal, O. Pandey et al., "Attribute-based encryption for fine-grained access control of encrypted data", In Proceedings of the 13th ACM conference on Computer and communications security, Pages: 89 - 98, 2006.

[21] R. Gray "Mobile Agents for Mobile Computing". Technical Report PCS-TR96-285, Department of Computer Science, Dartmouth College.

[22] R.S. Gray. "Agent Tcl: A Flexible and Secure Mobile Agent system". Proceedings of Fourth Annual Usenix Tcl/Tk Workshop, pp. 9-23, 1996

[23] D.Hankerson, A.J.Menezes and S.Vanstone. "Guide to elliptic curve cryptography". ISBN: 0-387-95273-X

[24] A.Harrington and C.Jensen. "Cryptographic Access Control in a Distributed File System". In Proceedings of the eighth ACM symposium on Access control models and technologies, Pages: 158 -165 ACM Press, 2003.

[25] D. Hearn, T.Wilkinson and S.Wiseman. "Trustworthy access control with untrustworthy web servers". In Proceedings of the 15th Annual Computer Security Applications Conference, page 12, IEEE Computer Society, 1999.

[26] Special Publication 800-88: Guidelines for Media Sanitization. NIST. September 2006. http://csrc.nist.gov/publications/nistpubs/800-88/NISTSP800-88_rev1.pdf. Retrieved 2007-12-08.

[27] JADE Security Add-On. http://jade.tilab.com/dl.php?file=securityAddOn-3.6.zip.

[28] Personal Health Information Protection Act (PHIPA 2004) http://www.health.gov.on.ca/english/public/legislation/bill_31/personal_info.htm.

[29] A. Joux. "A one round protocol for Tripartie Diffie-Hellman". In Proceedings of the 4th Algorithmic Number Theory Symposium, volume 1983 of LNCS, pages 385-394. Springer, 2000.

[30] G.Karjoth, D.B.Lange and M.Oshima. "A Security Model for Aglets," Mobile Agents and Security, Giovanni Vigna (Ed.), Springer-Verlag, pp. 188-205, 1998.

[31] M.Kasahara, K.Ohgishi and R.Sakai. "Cryptosystems based on pairing". In 2000 Symposium on Cryptography and Information Security –SCIS 2000, 2000

[32] N.Ksrnik and A.Tripathi. "Agent Server Architecture for the Ajanta Mobile-Agent System". In Proceedings of the 1998 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA ' 98), pp. 66-73, July 1998

[33] M.Luther. "Introduction to Identity-Based Encryption". ISBN: 9781596932388 9781596932395, 2008

[34] W.M.Farmer, J D.Guttmann and V. Swarup. "Security for Mobile Agents: Issues and Requirements". Proceedings of the 19th National Information Systems Security Conference, vol. 2, pp 591-597, 1996

[35] A.J.Menezes, T.Okamoto and S.A.Vanstone. "Reducing elliptic curve logarithm to logarithms in a finite field", IEEE Trans.Inf. Theory, 39, No.5 (1993) 1639-1646.

[36] W. Michalowski, D.O'Sullivan, S.Matwin, S.Wilk and K.Farion. "Designing an Agent to Support the Retrieval of Medical Evidence to Support Emergency Physician Decision Making at the Point of Care", CSM 2007: The 21st Workshop on Complex Systems Modeling, August 2007, Laxenburg, Austria.

[37] H. Peine and T.Stolpmann. "The architecture for the Ara Platform for Mobile Agents". In Proceedings of the First International Workshop on Mobile Agents (MA '97), vol. 1219 of Lecture Notes in Computer Science pp. 50-61, 1997.

[38] A. Sahai and B.Waters. "Fuzzy Identity Based Encryption". In Advances in Cryptology – Eurocrypt, volume 3494 of LNCS, pages 457–473. Springer, 2005.

[39] A.Shamir. "Identity-based cryptosystems and signature schemes". In advances in Cryptology-Crypto'84,volume 196 of LNCS, pages 47-53, Springer, 1984

[40] S.R.Tate and K.Xu. "Universally composable secure mobile agent computation". In Proceedings of the 7th International Conference on Information Security (ISC), pages 304–317, 2004.

[41] G.Vandana and S.R.Tate. "SAgent: a security framework for JADE". In Proceedings of the fifth international joint conference on Autonomous agents and multi agent systems Pages: 1116 – 1118, 2006

[42] E.R.Verheul. "Evidence that XTR is more secure than supergingular elliptic curve cryptosystem". In advances in Cryptology-Eurocrypt 2001. LNCS 2045,2001, 195-210 Springer-Verlag

[43] D. Wong, N.Pacoprek, T.Walsh, J.Dicelie, M.Yong and B.Peet. "Concordia: An infrastructure for Collaborating Mobile Agents", Proceedings of the First International Workshop on Mobile Agents (MA '97), vol. 1219 of Lecture Notes in Computer Science pp. 86-97, 1997

[44] A.Yao. "How to generate and exchange secrets". Proc. of the 27th IEEE Symposium on Foundations of Computer Science(FOCS), pages 162–167, 1986.