ENSURING DATA SECURITY AND INDIVIDUAL PRIVACY IN HEALTH CARE SYSTEMS

YANJIANG YANG

NATIONAL UNIVERSITY OF SINGAPORE

2006

ENSURING DATA SECURITY AND INDIVIDUAL PRIVACY IN HEALTH CARE SYSTEMS

YANJIANG YANG
(B.Eng. and M.Eng., Nanjing University of Aeronautics and Astronautics; M.Sc., National University of Singapore)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

NATIONAL UNIVERSITY OF SINGAPORE

2006

Acknowledgments

First and foremost, I wish to express my deepest gratitude to my supervisors Professor Beng Chin Ooi, Dr. Feng Bao, and Professor Robert H. Deng, for their profound guidance, advice and support that have made this thesis possible. I am fortunate enough to have all of them as my advisors, and I have greatly benefited from their exceptional insight, enthusiasm and experience in research.

I am deeply grateful to Professor Mohan S. Kankanhalli, Dr. Jianying Zhou, and Professor Kian-Lee Tan, who served as reviewers at different stages of my doctoral study. I would like to express my appreciation for their suggestions, comments, and time.

I would like to thank all my colleagues in the Infocomm Security department, Institute for Infocomm Research, and in School of Information Systems, Singapore Management University.

Finally, I would like to thank my wife and my parents for their love, encouragement, patience that helped me achieve this goal.

Table of Contents

Summary

Despite the great potential it promises in enhancing quality and reducing costs of care, information technology poses new threats to health data security and patient privacy. Our study in this dissertation thus focuses on technically addressing concerns of data security and especially individual privacy arising from current health care systems that represent a highly dynamic, distributed, and cooperative setting. In particular, we give a systematic study of the following typical yet closely related issues.

We first discuss user authentication techniques, building a unified trust infrastructure for health care organizations. User authentication is a fundamental and enabling service to achieve other aspects of data security within or beyond organizational boundaries. Discussions in this part thus lays a foundation for solving other data security and individual privacy issues in this dissertation and beyond. We suggest incorporating various user authentication techniques into a unified trust infrastructure. To that end, each organization establishes a *security manager* overseeing the organizational trust infrastructure and manages security related matters. Of particular interest is unifying password authentication into the trust infrastructure by a novel two-server password authentication model and scheme. The two-server system renders password authentication compatible with other authentication techniques, and also circumvents weaknesses inherent in the traditional password systems.

The next issue we study is to present a remote login scheme that allows users to

access a health care service in an anonymous manner. In other words, outside attackers cannot *link* different accesses by the same user. Our proposed scheme possesses many salient features, including resilience to DoS attacks. In later chapters, the anonymous login scheme and the user authentication techniques discussed earlier (e.g., password authentication) could be adapted for the purpose of entity authentication if necessary. However, as this is straightforward and orthogonal to the issues discussed thereof, we do not consider this aspect.

The scenario the anonymous login scheme deals with is by nature still at the level of individual organizations. We next explore a more complicated, inter-organizational procedure, medication prescription. We clarify and address privacy concerns of patients as well as doctors by proposing a smart card enabled electronic medication prescription system. Care is given to protect individual privacy while still enabling prescription data to be collected for research purposes. We also make the system more accord with real-world practices by implementing "delegation of signing" that allows patients to delegate their prescription signing capabilities to their guardians, etc.

The last topic we study in a broad sense continues the class of research on "achieving user privacy while enabling medical research" as the medication prescription system, but considers a quite different scenario: a health care organization (e.g., a hospital) outsources the health data in its repository to other organizations (e.g., a medical research institute). This actually involves "secondary" use of health data, which are an aggregation of medical records rather than individual records (the medication prescription system deals with individual records). Privacy protection therefore should be enforced at a level beyond individual data items, and the outsourcing organization has more

interests to be protected against the receiving organizations. In particular, ownership enforcement over the data in outsourcing is another issue to be addressed, in addition to the protection of individual privacy referred to in the data. We seamlessly combine *binning* and *digital watermarking* to attain the dual goals of privacy and copyright protection. Our binning method allows for a broader concept of generalization, and our watermarking algorithm is a hierarchical scheme resilient to the specific *generalization attack*, as well as other attacks common to database watermarking. The experimental results demonstrate the robustness of our techniques.

List of Tables

List of Figures

Abbreviation List

AES	Advanced Encryption Standard
CA	Certification Authority
CRL	Certificate Revocation List
DAC	Discretionary Access Control
DDH	Decisional Diffie-Hellma
DH	Diffie-Hellman
DICOM	Digital Imaging and Communications in Medicine
DLP	Discrete Logarithm Problem
DoS	Denial-of-Service
EMR	Electronic Medical Record
GP	General Practitioners
HIPAA	Health Insurance Portability and Accountability Act
HL7	Health Level Seven
HMO	Health Maintenance Organization
IDS	Integrated Delivery System
MAC	Mandatory Access Control
PAKE	Password-only Authenticated Key Exchange
PBM	Pharmacy Benefits Management System
PDA	Personal Digital Assistance
PKI	Public Key Infrastructure
PVD	Password Verification Data
RABC	Role Based Access Control
RDBMS	Relational Database Management System
RSA	R. Rivest, A. Shamir and L. Adleman
SSL	Secure Socket Layer
TTP	Trusted Third Party
WWW	World Wide Web

CHAPTER 1

Introduction

1.1 Motivation

Information technology becomes increasingly essential to health care, enabling the health care industry to improve the quality of care provision while at the same time reducing its cost. This trend is clearly witnessed by the fact that more and more health care organizations are developing electronic medical records (EMRs) for facilitating clinical practice, setting up internal networks for sharing information and simplifying administrative and billing processes, utilizing public networks especially Internet for enabling inter-organizational collaborations of care, reimbursement, benefits management, and research. The application of information technology to health care both drives and is driven by structural changes of the health care industry and its methods of care. Take U.S. for example, during the past few years, the health care industry has seen the following significant changes [53, 59].

- Consolidation of care providers that serve different aspects of the care continuum, e.g., hospitals and primary care clinics, into integrated deliver systems (IDSs). IDSs may also include financing services that offer health plans and pay for care. The rapid growth of IDSs is due largely to the promises of cost savings and expansions of market share through consolidation and federation, and the improvements in the quality of care by a continuum of time of care management. IDSs entail a significant increase in the use of information technology to store, analyze and share health data within and possibly beyond the limit of individual IDSs. IDSs now rapidly become the primary means of care delivery.

- Rise of managed care, such as health maintenance organizations (HMOs), has greatly altered the practice of medicine and created new demands for information. Managed care uses capitation systems to pay for health care and manage risk, in contrast to traditional ways of insurance where care providers or patients are reimbursed for services they offered or received. In a capitation system, care providers get reimbursed based on the number of patients enrolled in their care rather than on the services rendered. Meanwhile, managed care providers involve extensive examination of aggregate data to define optimal approaches to the management of chronic diseases; introducing increasingly sophisticated approaches to manage care of groups of patients with similar health problems; analyzing the use of medical resources such as medications, specialists, and surgical procedures. Managed care has contributed to a shift in the view of medical care from mostly an *art* based on clinical judgement to mostly a *science* based on empirical data.
- New entrants that collect and consume health information. These organizations typically provide products and services to the health care industry and have developed significant business interests that involve the collection and analysis of health data. Medical data surgical suppliers, pharmaceutical companies, and reference laboratories are such organizations. Furthermore, existing players in the industry are expanding their roles. For example, insurance companies establish

their own provider networks, and care providers begin to set foot in administration and financing of care.

By and large, these changes have led to a tremendous increase in the collection and use of patient health data and in the sharing of health data across organizational boundaries. A central enabling element to the above trends of integrated functions and managed care is the development of Electronic Medical Records (EMRs).

EMRs provide comprehensive and accurate information concerning patients' medical histories, health problems, laboratory results, therapeutic procedures, medications, account management and billing, etc. Over time, the content of EMRs is anticipated to expand beyond that of paper records and include medical imagery and telematic video [53]. EMRs offer many advantages over traditional paper records. The primary benefit of using electronic records is efficient and flexible access. For example, EMRs allow multiple users simultaneous access to the information from a variety of locations; with EMRs, fine-grained access is possible in the sense that access can be limited to just the portion of the record that is pertinent to the user. Electronic data can also be used to accomplish tasks that are not possible in the paper format. For example, electronic records can be organized, displayed, and manipulated in a variety of different ways that are tailored to particular clinical needs. This in turn enables the capabilities of real-time quality assurance, decision support systems, event monitors, and availability at the point of care. Electronic records also promise the improvement of clinical research based on extensive analysis of clinical data [12, 43, 176].

Clearly, information technology has revolutionized health care into a setting of increasing computerization and networking. Nevertheless, the wide and extensive use of information technology poses new threats to health data security and individual privacy contained in the data. Traditional paper records had a physical embodiment, were awkward to copy, and were accessible only from central repositories. The difficulty of moving information increased dramatically with the volume of records being transferred. Computerization and networking have changed this situation radically. EMRs have no physical embodiment, are easily copied, and are accessible from multiple points of access. Large numbers of records can be transferred as easily as a single one. The existence of networks and especially Internet makes data transfer across administrative, legal, and national jurisdictions to a maximum ease. However, it is obvious that the advantages offered by EMRs and networking can be adversely exploited for purposes of compromising security of health data.

EMRs also raise the possibility that accurate and complete composite pictures of individuals can be more easily drawn. As a result, people would reasonably raise concerns about the aggregate even if they had no concerns about any single data element. In an electronic system, large scales of data retrieval and data aggregation can be accomplished almost instantaneously and invisibly. Moreover, any such aggregated database itself might become an interesting target for those seeking information. The emergence of new information processing tools e.g., data mining [14], that are widely used for research purposes [43, 176] signifies the emerging challenges in keeping individual privacy in health care systems, where data outsourcing and secondary use of data are becoming common now.

It is now clear that information technology has on the one hand greatly benefited health care by changing its practice and methods of care, while on the other put data security and individual privacy in an ever more vulnerable state. This thus motivates the need for protection of health data. We next discuss the significance in maintaining data security and individual privacy in health care.

1.1.1 Why Security and Privacy Matters

Health data are in nature private and sensitive, and keeping patient privacy is quite relevant to the fundamental principle of respecting human right in a civilized society. In practice, compromise of data security and individual privacy may result in varying sequences to individuals, ranging from inconvenience to ruin. For instance, inappropriate disclosure of health information could harm patient's economic or social interests, such as causing social embarrassment [59] and affecting employment and health insurance acquisition [106,122,145]; if patients believe their information cannot be kept confidential, they would be reluctant to share health information with their doctors, which results in reduced quality of care; the corruption of health data might mislead doctors to wrong treatment for patients, thereby damaging the patients [18,151].

As far as care providers are concerned, the ethical and professional obligation for protecting patient privacy has long been well recognized. Since the fourth century B.C., physicians have abided by the Hippocratic oath in keeping secret patient information they learnt in the course of care: *Whatsoever I shall see or hear in the course of my dealing with men, if it be what should not be published abroad, I will never divulge, holding such things to be holy secrets.* Over the centuries, the bound upon the health care community by ethical and professional obligation has never been weakened, and new codes of ethics adaptable to the dramatic changing health care setting are continually under review [105]. On the other hand, realizing their health information may be at stake in today's digital era, public attention to the security and privacy of health information is at an all-time high. According to a latest survey [174] conducted by the Medical Record Institute, U.S., up to 76.9% of the respondents worried about security of patient data, and 60.2% prioritized privacy breaches by authorized or unauthorized users as a "major concern" regarding data security. Health care organizations hold the responsibility to mitigate public worries, as maintaining their patients' privacy is a matter of trust and an important factor in sustaining a positive public reputation. Privacy breaches may erode public confidence on care providers, and the industry as a business would be harmed.

Furthermore, protection of health data and individual privacy is now quite under the jurisdiction of laws around the globe, going far beyond the scope of ethical, professional responsibilities and business interests. For example, in U.S. there are both federal and state laws and regulations on the protection of health information [180], among which the Health Insurance Portability and Accountability Act (HIPAA) [86] represents the latest and the most comprehensive drive for security in health care; in Europe, European Union issued the Recommendation R(75) [147] and Privacy Directive [144], etc., and each member country has its own laws and ethical codes as well, such as the Health and Social Care Act 2001 in UK; other countries have similar laws, regulations, and ethical codes: Singapore has the Medical Ethics & Health Law [167], South Korea has act regulating the protection of personal information maintained by public agencies [108] and Japan has the Data Protection Bill [98]. Under legal mandates, health care providers responsible for privacy breaches will be, and have been sued and subjected to administrative sanctions [59].

As made it clear, ensuring data security and individual privacy favors not the sole interest of patients, but also that of the overall health care industry; protection of health data concerns not only the good ethical and professional faith, but also compulsory compliance with laws, regulations and codes of ethics.

1.1.2 Challenges in Protection of Health Data

As stated earlier, health care is a setting of federation and consolidation of various organizations with interleaving interests, security and privacy concerns thus arise from within individual organizations, across integrated delivery systems, between and among providers, payers, and secondary users [53]. We discuss the challenges in protection of health data in such a highly complex setting from the following perspectives.

At the policy level, great differences exist among distinct stakeholders as to what constitutes valid use of the health information. No consensus exists across the health care community regarding the legitimacy of each stakeholder's demand for health information. This lack of consensus differentiates health care domain from military and financial sectors where a general consensus on information policy exists [8]. Consequently, consistent policies synchronizing interests of various stakeholders in the federation of organizations are quite challenging. Even at the level of individual organizations, policy establishment is also very difficult. A wide range of context factors complicate access management in health care. They include conflict interests between patients and care providers over the security and privacy of data; different perspectives on the access issues by different stakeholders [141]; diversity in health care business models and frequent changes of health care environment [29]; the role users' responsibility plays in access control [179]; different contextual elements of access, such as time, location, etc [11]; the involvement of non-medical parties, such as medical research, employers, clearing houses; emergency access of health data.

From a technical perspective, data protection should be enforced upon data in storage, data in transmission, data in business transactions, and data in sharing. As such, there exists no one-size-fits-all solution for the protection of health data [73, 175], and diversity of the health care setting entails methods that are tailored to the specific scenarios and needs. For example, in the process of medication prescription, individual privacy includes not only patient's privacy but also doctor's privacy stake, and their privacy concerns vary with respect to different parties such as pharmacy and insurer. A sound solution to medication prescription has to address every aspect of the privacy issues. As an another example, [148] empirically demonstrated the failure of deploying firewall without attuning to the unique requirements of a health care application. In health care, data protection techniques and solutions must take into consideration of the different types of modalities (e.g., text, image, audio) contained in the health data, as well as the various facilities (e.g., Internet, wireless networks, workstations, servers) upon which health care applications are built. Moreover, it is prudent to attune security solutions to the real-world medical practice. Otherwise, significant overheads and obstacles would be incurred upon normal working practice, and it is also being seen as a serious assault on professional independence [13]. Protection solutions in health care should also integrate and keep compatibility with the legacy systems that have consumed large amounts of money, and are currently providing for the smooth functioning of routine tasks. Finally, it is important to notice that information sharing that leads to secondary use of health data goes beyond simple exchanges of data among organizations, yielding new security issues in current health care systems.

With appropriate policies and techniques in place, organizations may still have operational handicap in enforcing security [8]. First, security practice has its its uniqueness in health care. Unauthorized accesses to data in military and financial domains are likely to be used for criminal purposes, e.g., the spies steal military or financial secrets. With health information, such breaches and uses may be more insidious, whereas the damages are less overt. Managing staffs in both military and financial organizations are given strong liability to curb criminal use of the housed data; breaches are often followed by punishment. In contrast, security breaches in health care organizations are less likely to be made public, and the public normally presumes the high ethical standards upon health care personnel are enough of a deterrent to the data misuse, which turns out not to be the case in practice. Second, security deployments in health care industry lack market incentives. Patients generally select care providers and health plans for reasons other than their ability to protect patient information. The fact however is that information security has proven itself to be more of business policies and procedures that must be managed from a business perspective [30]. The lack of market incentives, together with the impropriate views such as investing in security decreases performance and increases costs, would hinder active executive involvement. Third, most health care professionals do not keep pace with the advances of information technology, and they often lack awareness and training in security enforcement. Human factor can constitute the weakest link in the chain of security. User awareness promotion and training has been repeatedly outlined in virtually any guideline on health data security (see for example, [30, 52, 57, 161–163]).

From a legal perspective, security solutions in health care must comply with legislation. No industry is more challenging for technicians than health care in this aspect, and they have to navigate a set of laws, regulations and codes of ethics in an attempt to find satisfactory solutions [152]. The adopted security solutions must at a minimum meet the stipulations of health laws, standards, codes of ethics and other relevant laws and regulations.

To summarize, protection of health data in health care systems is not purely a technical issue, with social and organizational factors also playing a major part [6]. And it is important to bear in mind that technology alone cannot safeguard health data, and sound solutions require balanced implementation of sound security policy, good system administration practice, proper management and use of technology, and strict accordance with law regulations.

1.2 Scope of the Research

We have seen that information technology has posed considerable threats to health data, and protection of health data and individual privacy is of great significance but challenging in current health care systems. This motivates and justifies our study on ensuring data security and individual privacy in this dissertation. Before discussing our contributions, we first see general security requirements for health care systems.

1.2.1 Security Requirements for Health Care Systems

To make our discussions more concrete and clearer, we derive security requirements upon health care systems from a typical setting in health care as shown in Figure 1.1, which includes several typical parties involved in care, together with the data flows among them. We stress that the intent of this figure is not (and in fact impossible) to cover the whole health care setting that includes all parties and all data flows, but to convey some characteristic aspects of practical health care systems. Moreover, extending this representative system to more complex scenarios by integrating other parties, functions and data follows is not difficult.

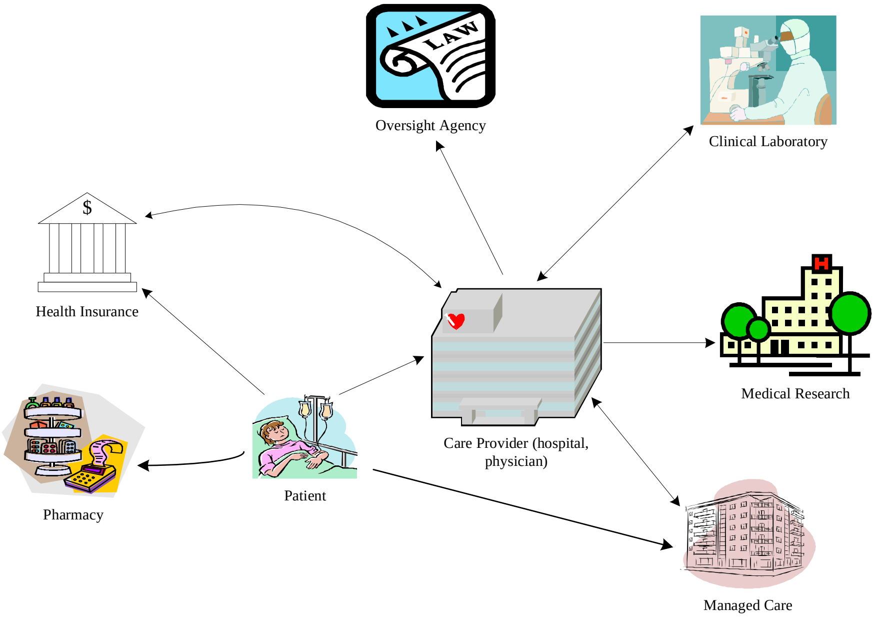


Figure 1.1: A Typical Health Care Setting.

In this setting, patient Alice chooses her *care provider* (e.g., physician and hospital), and naturally, the care provider hosts her medical records. The care provider may sometimes need to send, e.g., blood samples, to an outside *clinical laboratory* for test and analysis. As a result, the laboratory will retain a record of the test. When Alice enrolls in a health benefits plan, the *health insurance company* obtains her health data. One day Alice is ill, then she visits her physician who prescribes medication for her. Alice files the prescription at the local *pharmacy*, so the pharmacy keeps parts of her health data. To enhance care quality and efficacy, the care provider may cooperate with *managed care* for cost-effective research (or the care provider itself belongs to the managed care). Frequently, *Medical researchers* acquire health data for the purpose of medical research after gaining approval of his institutional review board and permission of Alice. Upon court order, the care provider bears the responsibility to provide patient data to the *governmental oversight agency* for investigation, or for checks on the record-keeping procedure at the provider organization. This is a very brief description of the system. Moreover, in many cases, how and where the patient data are collected may vary, quite dependent on factors such as what health plan Alice enrols in, and how the parties associate with one another.

A main characteristic represented by Figure 1.1 is that health care is a highly networking and cooperative setting, and health data are distributed across various places. Based on this setting, we derive the following general security requirements for health care systems.

Confidentiality: Confidentiality consists of the fundamental part of data security, referring to the holding of information from inappropriate disclosure. The demand for data confidentiality in health care is clear since health data are sensitive and private in nature. Achieving data confidentiality involves protection of health data in repository by means of *access control* within individual organizations, as well as protection of data in transmission across organizational boundaries.

Privacy: Privacy is a term often confused with confidentiality, and they are often used exchangeably. But in a strict sense, they have differences. In simple terms, privacy refers to an individual's desire and right to be left alone and to be protected against physical and psychological invasion or the misuse of their property. In contrast, confidentiality concerns protecting against theft, disclosure, or/and improper use, and data must be disseminated only to authorized individuals or organizations with a need to know. Let us see an example: if an organization surreptitiously collects patient information for marketing purposes via the Internet, it is intruding an individual's privacy; if a doctor disclose a patient's records to an external doctor without permission, he is violating confidentiality.

For the purpose of discussions in this dissertation, we draw a easy distinction between confidentiality and privacy as follows: (individual) privacy clearly relates to (data) confidentiality, and the simplest way to achieve individual privacy is to maintain data confidentiality, i.e., not to disclose the data in question whatsoever. But achieving individual privacy involves more than simply keeping data confidential. Consider a scenario in Figure 1.1 where the health care provider shares patient records with the medical research for investigating the long-term effects of certain medications. The care provider wants to keep patient privacy, so de-identified data are transferred to the medical research. Data confidentiality in this scenario is not a concern with respect to the medical research, since the medical research gets the data; in contrast, individual privacy involves preventing the medical research from inferring some useful information on patients from the de-identified data.

In practical applications, *anonymity* and *unlinkability* are two important properties of (individual) privacy, where anonymity refers to the prevention of disclosing information that leads to the identification of users (anonymity has no identifying information and hence provides privacy), and unlinkability requires that data of the same user cannot be recognized as such. Unlinkability amounts to a form of strong user privacy.

Health care professionals also have privacy concerns in health care systems. For example, doctors' prescription patterns are contained in their prescription histories, which could be taken advantage of by the corresponding care provider to act against the doctors themselves [8]. Therefore, individual privacy in health care systems includes both patient privacy and doctor privacy. Moreover, as secondary use of health data for purposes such as clinical research and cost-effective research becomes increasingly common, ensuring individual privacy is of urgency yet complex.

Integrity: Integrity refers to the assurance of information being kept intact. Significance of maintaining integrity of health data is clear: corruption of health data could delay patient treatment for lack of right information or mislead the health care professionals to give wrong treatment. It is prudent practice to check data integrity before use of the data, be they fetched from local storage or transferred from other locations.

Authentication: Authentication comprises *user authentication* and *data authentication.* User authentication is to establish the validity of a claimed identity, while data authentication involves verifying the integrity and authenticity of the data (authenticity can be understood as the confidence in the validity of a transmission, a piece of data, or data originator). User authentication is a premise for enforcing access control over health data in repositories. Data authentication is by necessity a crucial step in ensuring that data originate from the claimed party and have been kept unmodified.

Availability: Availability refers to the fact that upon request, a system provides data (or service) to the authorized users in a timely fashion. Disruption of data availability can cause catastrophic consequences [10] in health care systems. Furthermore, data availability is of paramount importance in the event of emergency, in which case data availability must override other concerns.

Accountability: Accountability is the assurance that any access to data is recorded and can be traced. Accountability offers one of the strongest deterrents to data abuses by both insiders and outsiders. Accountability is normally achieved by auditing trails that closely couple with access control mechanisms. The content of the auditing trails normally contain details about data access, typically including the identity of the requester, the data and the time of the access, the source and the destination of the access and a reason for access.

Non-repudiation: Non-repudiation is the cryptographic service that legally prevents the originator of a message from denying authorship at a later date [164]. Non-repudiation has been a quite useful property in e-commerce, and we believe it is also critical for inter-organizational transactions in health care. For example, [23] gave a health care application that implemented non-repudiation.

Rights Enforcement: As already stated, secondary use of health data is quite common and essential in current health care practice (e.g., in Figure 1.1, care provider shares patient data with medical research or with managed care). However, when health data are given to the secondary users, care provider will lose complete control over the data. A concern of redistribution of the data by the secondary users thus arises. For one thing, health data are an important asset to care provider that has collected and compiled the data, and it is thus important for care provider to assert data ownership. For the other, in case of data redistribution leading to inappropriate information disclosure, rights enforcement tools can help trace the liable parties. While agreements are normally signed by the secondary users on the use of the shared data (e.g., they promise not to illegally redistribute the data), technical means is a more effective deterrent. Note that traditional security mechanisms such as access control and encryption are completely ineffective in controlling health data in secondary use. As a matter of fact, health care community now has growing awareness on the need of right enforcement (see e.g., [49, 53, 104, 183]), and it is suggested that effective right enforcement over systemic data flows is among the primary unresolved technical problems in health care domain [53].

These are the common and general security requirements for health care systems, although they may need to be refined when considering particular applications, scenarios and contexts. While construction of a secure health care system should consider all these requirements (each may require considerable work), our main focus in this dissertation is on ensuring individual privacy.

1.2.2 Our Contributions

Health care community has long held the obligations for ensuring data security and individual (patient) privacy, and as a result, considerable effort has been dedicated to this subject. Most of the past investigations in general concentrated on either establishing guidelines and policies on regulating health care professionals and practices (e.g., [10, 17,40,52,53,59,93,127,146,161–163,179]), or constructing end-to-end secure health care systems with off-the-shelf security tools from a more engineering-inclined perspective (e.g., [19,33,35,36,39,111,125,130,152]). Work in the former class normally examined insufficiencies in the protection of health data by health care professionals and care organizations, and then came up with recommendations, guidelines and policies towards correct health care practice. In contrast, work in the latter class explored applying well established security primitives and techniques to health care systems, for example, using encryption to secure data in transmission, or developing access control mechanisms and deploying firewall to safeguard data in storage. Limitations of these existing methods include (1) most of them constructed "secure systems" that keep *confidentiality* of health data, that is they concentrated mainly on making data secure in the local storage and in transmission; (2) they seldom considered protecting health data in secondary use where care providers lose control over the data. This aspect becomes clearer as secondary use of health data becomes increasingly important and common in current and future health care systems; (3) few work addressed emerging issues and concerns such as rights enforcement and *strong* individual privacy. Take individual privacy for example, patients are increasingly concerned about individual privacy in care, as with in e-commerce. However, health care community is slow in responding to the increasing demand for strong individual privacy such as *unlinkability* in health care transactions; (4) some of the existing proposals might not endeavor to respect the real-would practice when achieving protection objectives; (5) compliance with law regulations was not seriously taken into consideration by many of the existing work. But things changed dramatically now: for example, in U.S. as HIPAA takes effect, organizations failing to meet law stipulations would incur legal action.

The scope of this dissertation is concerned with a systematic study on techniques to solve data security and especially individual privacy issues arising from health care systems, represented by the setting in Figure 1.1. The intent of our study is not to construct a complete health care system, but to address several typical and closely related issues, and the techniques we propose can be readily integrated into any practical system. We focus on emerging issues in current and future health care systems, and our design takes into consideration the compliance with law regulations (e.g., HIPAA [86]). For the latter, throughout the dissertation we assume that each patient (or physician) has a personal smart card at her disposal, and each care organization has a security manager (we shall discuss security manager in Chapter 3). Note that HIPAA endorses the use of smart card and the establishment of security manager (where it is termed privacy officer) in health care systems.

In particular, the following issues will be studied in this dissertation.

- *Building a unified trust infrastructure for health care organizations.*

  User authentication is a fundamental part of data security, and is an enabling service for achieving other aspects of data security within or beyond organizational boundaries. As signified by Figure 1.1, health care systems represent a highly dynamic and complex setting, various user authentication techniques are thus necessary for satisfying varying demands of organizational and inter-organizational applications. We suggest each care organization build a unified trust infrastructure that accommodates various user authentication techniques and modes, e.g., short password, identity certificate, attribute-based certificate, anonymous credential, and group signature. To that end, each organization establishes a *security manager* that manages the organizational trust infrastructure and handles security related matters. Our focus is to unify password authentication with other authentication techniques that can directly exploit the security manager as the CA (Certification Authority) or the TTP (Trusted Third Party). Our solution is a novel two-server password authentication model and scheme that enlists the security manager for operating a back-end authentication server. In addition to rendering password authentication compatible with other authentication techniques in the organizational trust infrastructure, the two-server password authentication system also circumvents weaknesses inherent in the traditional password systems, e.g., off-line dictionary attacks against the server password database. We remark that the establishment of organizational trust infrastructure in individual care organizations enables to solve other data security and individual privacy issues beyond this dissertation.

- *Anonymous remote login for health care services.*

  With the organizational trust infrastructure in place, we are ready to solve security issues beyond organizational boundaries. We know that an immediate effect of the application of information technology, especially EMRs and networking, in health care is the enabling of care organizations to allow users (e.g., physicians and patients) to access clinical services and data from off-site locations. However, sensitive information pertaining to users such as individual preferences, life styles, health conditions is conveyed from the services they are accessing. For example, if a user frequently visits the website of a dental service, most probably the user has dental problem. Furthermore, when widely collected and compiled, user information of such a kind can be used by some organizations for marketing purposes, etc [22]. In light of this, we propose an anonymous login scheme that enables a care organization to provide anonymous remote login service to its affiliated physicians and patients. In other words, the proposed anonymous login scheme hides users' access patterns in the process of remote login, so that accesses by the same user cannot be linked. The scheme has many salient features, including resilience to DoS attacks, which is important yet hard to achieve in anonymous systems. Anonymous login is of particular interest and importance as long as users are concerned about their privacy involved in the login process, and are willing to attain strong user privacy.

We stress that in later chapters, the anonymous login scheme and the user authentication techniques discussed earlier (e.g., two-server password authentication) could be adapted or extended for the purpose of establishing trust among parties that involve in inter-organizational transactions if they need to authenticate each other and/or set up a secret channel. However, as this may be straightforward and orthogonal to the issues in question there, we often do not explicitly discuss this aspect.

- *Privacy preserving medication prescription.*

  The scenario the anonymous login scheme considered is essentially still at the level of individual organizations. We next move to a more complex, inter-organizational process, namely, medication prescription. Medication prescription is a routine process in health care systems. The following facts suggest protection of individual privacy involved in medication prescription needs special treatment. First, the involvement of diverse parties, especially non-medical parties in the process complicates the protection of prescription data. Second, both patients and doctors have privacy stakes in medication prescription, and the privacy concerns have distinct implications with respect to different parties. Third, medication prescription should not proceed in a truly anonymous manner: certain involved parties need to conduct useful research on the basis of aggregation of prescription data; prescription data has to also be identifiable in some extreme circumstances, e.g., under the court order for inspection or for assignment of liability. Another important issue in medication prescription needs to be addressed is the delegation of prescription filing capability by patients to other people. This feature accords with a common practice in medication prescription that others instead of the patients themselves collect the prescribed medicine. They may be a patient's guardians, relatives, or friends who accompany the patient to visit the doctor.

To address all these issues, we propose a smart card enabled electronic medication prescription system. Smart cards carried by patients play an important role: for one thing, smart card is implemented to be a portable repository carrying up-to-date personal medical records and insurance information, providing doctors instant data access crucial to the process of diagnosis and medicine prescription; for the other, with the private signing key being stored inside, smart card enables a patient to sign electronically the prescription pad, declaring her acceptance of the prescription. A strong proxy signature scheme achieving mutual agreements on delegation is proposed to implement the delegation functionality.

- *Privacy preserving and right enforcement of health data in outsourcing.*

  A main objective of the medication prescription system is to ensure individual privacy (of doctors and patients), while at the same time enabling relevant parties to collect prescription data so as to conduct research based on the statistical analysis of these data. We continue this line of study on "achieving user privacy while enabling medical research", but consider a quite different scenario: a health care organization (e.g., a hospital) outsources the health data in its repository to other organizations (e.g., medical research institute). This actually involves "secondary" use of health data. As we already made it clear, the demand for such kind of secondary use of health data is increasing steadily for purposes of clinical research and cost-effective research, which are essential in the provision of better quality care. The health data to be outsourced are an aggregation of medical records rather than individual records. Privacy protection therefore should be enforced upon beyond individual data items, and the outsourcing organization has more interests to be protected against the receiving organizations. In particular, besides the protection of individual privacy referred to in the outsourced data, copyright (ownership) enforcement over the data is another issue to be addressed. We present a unified framework that seamlessly combines techniques of binning and digital watermarking to attain the dual goals of privacy and copyright protection. Our binning method is built upon an earlier approach of generalization and suppression by allowing a broader concept of generalization. To ensure data usefulness, we propose constraining binning by usage metrics that define maximal allowable information loss, and the metrics can be enforced off-line. Our watermarking algorithm watermarks the binned data in a hierarchical manner by exploiting the very nature of the data. The method is resilient to the *generalization attack* that is specific to the binned data, as well as other attacks intended to destroy the inserted mark. We implemented the techniques, and the tests show promising experimental results.

1.3 Organization of the Dissertation

The rest of this dissertation is organized as follows. In chapter 2, we review related work that demonstrates the status quo of security implementation in health care systems. We also give a brief survey on access control in health care in a separate subsection, considering access control is an extensively studied topic in literature; moreover, to facilitate the development of practical access control for health care applications, we list many features that are necessary for health care systems and thus should be incorporated in the basic role based access control model.

In Chapter 3, we discuss user authentication in health care systems, and suggest building a unified trust infrastructure for health care organizations that accommodates various user authentication techniques and modes, e.g., short password, identity certificate, attribute-based certificate, anonymous credential, and group signature. To make password authentication compatible with other authentication techniques and modes, we propose a novel two-server password authentication system.

In Chapter 4, we propose an anonymous login scheme that provides users (physicians and patients) remote access of clinical services and data in an anonymous manner. The proposed scheme possesses a salient feature that different login sessions by the same user cannot be linked by outside attachers, thereby achieving strong user privacy.

In Chapter 5, we present a smart card enabled privacy preserving medication prescription system. We analyze different implications of patient privacy as well as doctor privacy with respect to different parties, and accordingly address these privacy concerns. To well accord with the real world practice, we propose a strong proxy signature scheme for the purpose of implementing "delegation of signing" in medication prescription. Smart card is heavily used in this proposal: on the one hand, smart card is enlisted as a portable repository housing up-to-date personal medical records and insurance information; on the other hand, smart card serves as a tamper resistant device storing inside private signing keys, enabling signing of medication prescription pads.

In Chapter 6, we investigate preserving of individual privacy and ownership of health data in outsourcing. In particular, we present a unified framework that seamlessly combines techniques of binning and digital watermarking to attain the dual goals of privacy and copyright protection. Our binning method extends an earlier approach of generalization and suppression, allowing a broader concept of generalization. Our watermarking algorithm is a hierarchical scheme by exploiting the very nature of the binned data. It is resilient to many malicious attacks intended to destroy the inserted mark, including *generalization attack* that is specific to our scenario. We conducted extensive experiments on the proposed techniques, and promising results were obtained.

Finally, Chapter 7 summarizes the dissertation.

CHAPTER 2

Related Work

Due to the very nature of health data, health care community has long held the professional and ethical obligations for ensuring data security and patient privacy. As the use of information technology becomes increasingly prevalent in health care domain, considerable effort has been invested to studying security issues in health care information systems. Past studies in general fall into two classes: the first class focused on establishing guidelines and policies on regulating professionals and care practices, see e.g., [10, 17, 40, 52, 53, 59, 93, 127, 146, 161–163, 179]; the other class studied security solutions at a technical level. Since the latter class is more relevant to our work in this dissertation, we thus in what follows choose to give a brief introduction to the work in this class, which indicates the status quo of security implementation in health care systems. In addition, considering access control has bee an extensively studied topic in literature, we review access control in health care systems in a separate subsection for clarity reasons, and list many features that are useful and thus should be implemented in role based access control for health care applications. We believe these features can be a good starting point for the development of a practical access control mechanism for health care systems.

2.1 Security Implementation in Health Care

Medical images represent a main data modality in health care systems. Many applications explored securely accessing or transferring medical images. [87] reported a simple system designed to transmit pictures and radiographs of a severely fractured ankle over WWW (world wide web) for viewing by a consultant, followed by recommended treatment. Security however is not strictly strengthened in this system and only "password" is needed to access the medical images. Similarly, the system in [85] is also intended to deliver medical images via WWW. However, security is much more strengthened, e.g., a proxy server and a firewall is employed to safeguard health data from external attacks, and SSL is taken to protect data flow.

To facilitate health care application development over WWW, it is more and more accepted that proven middleware technology such as CORBA, structured representation technology such as XML, in combination with established health care standards must collaborate and integrate seamlessly. [74] examined systematically the efforts in developing component-based standards specific to health care environment that can fit finely with existing health care standards such as HL7 [83], and DICOM [58]: for example, OMG has developed COBARmed [51]; Microsoft's Healthcare Users Group commits itself to the development of ActiveX-based implementation of HL7 messaging objects [3]; XML is introduced to HL7 to enable the latter to take a more object-oriented view [84]. The system in [24] is an example making use of CORBA to construct an open, streamlined, automated, monitored platform to distribute clinical images in an environment of a large consortium of hospitals. In [26], CORBA and OLE were deployed into health care applications so as to evaluate their capabilities in developing distributed applications: Patient record data object providing a consistent specification for accessing the data was created by CORBA. Two clients of the data object were developed. One, using OLE, illustrates access information within the organization. The other, developed as a WWW browser, demonstrates access information from outside. [29] identified the main issues in constructing by CORBA compatible component-based technology a security environment where a user will be viewed the same across the heterogeneous systems, and access control decisions will be consistent across all components of the environment. [39] developed a toolset using object-oriented techniques including the unified modelling language (UML) to facilitate the different users' views for security analysis and design of health care information systems. The proposals we come up with in this dissertation are independent of the techniques and standards at the implementation level.

Legacy systems that are currently providing for the smooth functioning of their systems contain large prior investments; therefore it is necessary to integrate them into the new systems. The integration process however is difficult due to the incompatible format used and the facts that security factors are not taken into consideration in the time of their deployment. The solution strategy suggested in [103] was that information is communicated using messaging standards such as HL7, and archived and used in such ways that can exploit the Internet and distributed object technologies. An example is to develop and use a HL7-to-CORBA gateway. Similar approach was also used in [102] to adapt the legacy system to the new WWW environment. It appeared that hiding legacy systems behind "brokers" is a practically economical solution. The "brokers" act as an intermediary to support interoperability, implement security services at the application level.

The MIPA project (Medical Information Privacy Assurance) [2] was aimed to develop privacy technology and privacy-protecting infrastructures to facilitate the development of a uniform health care information system so that individuals can actively protect their personal information. The project contained three systems: an e-prescription system (which will be reviewed shortly), a credential-transfer system and a centralized anonymous repository for medical records. The credential-transfer system allowed for the construction of trust among health care organizations and their external business associates on the protection of patients' health data. Users use pseudonyms rather than anonymous transactions when they contact with organizations. The centralized, anonymous repository provided a Centralized Medical Database (CMDB), where strong user authentication and audit records of accesses are powerful abuse deterrents. The patients can specify who may read/update their medical records, and for what duration. So, in compliance with the HIPAA regulations, patients have control over their medical records. Anyone accesses the data must get consent from the patients. Exceptional cases, e.g., FBI agents access the data holding court order, were accommodated in this system. The flavor of the MIPA project is much similar to that in this dissertation, and most of the issues we study are independent of those in the MIPA project.

With HIPAA taking effect, health care community is forced to comply with the legal regulations on protection of health data. [152] systematically studied how PKI (Public Key Infrastructure) is used by health care organizations to comply with HIPAA. It also illustrated using PKI for important business solutions with the help of detailed case studies in other sectors such as financial, government, and consumer industries.

It highlighted how to meet domestic and international regulations for corporate-level and government-level standards on security and privacy. We believe PKI is an essential element for trust establishment in health care environment, whereas we discuss to include PKI as a part in the organizational trust infrastructure, which also incorporates may other user authentication techniques and modes (see Chapter 3).

Medication prescription is a typical routine service in health care systems, involving multiple parties (some are non-medical organizations), and these participants have distinct privacy concerns. Several work looked into addressing individual privacy in the process of medication prescription. The work in [136] was aimed at protecting doctors' identities in the prescription pads while at the same time allowing data to be aggregated for the purposes of research and statistical analysis. The enlisted method is presenting prescription data in two distinct batches: one batch includes prescription information with scrambled doctor references and the other batch contains the scrambled doctor references and the doctor information. A trusted third party is involved in the process so the data collector is not able to identify doctors who did not agree to being identified. The first batch is encrypted with a public encryption key of the data collector and the second batch is encrypted with a public key of the third party; both batches are sent to the respective parties. Only the third party, possessing the corresponding private key for decryption, can then recover the second batch data in the readable form. Analogously only the data collector can decrypt the first batch of data. The anonymous E-prescription system in [2, 8] sought to achieve similar objectives of protecting individual privacy while enabling useful research, but using quite different approaches. In particular, patient privacy is reserved by each patient applying for a pseudonym from the insurer and signing the prescription under the pseudonym, and anonymity revocation is accomplished by the insurer; anonymity of doctors is achieved by each doctor joining a group and then issuing group signatures on the prescription pads by using a group signature scheme (e.g., [1,47]). Another work relates to medication prescription is [132], which presented a clearing scheme for the Germany health care system, addressing the privacy issues among various parties such as physician, insurers, pharmacies, etc., in an overall context of medical processes. We also address individual privacy involved in medication prescription in Chapter 5, based on the above work and especially on [8]. However, two main features distinguish our system from others: first, we introduce smart card into our system, acting as both a repository housing latest personal health and insurance data and a prescription signing device; second, we implement *delegation of signing*, allowing patients to delegate their prescription signing capabilities to other people. This feature is of particular interest to e.g., disabled patients.

Finally, to get a complete picture on security implementation in health care systems, we next introduce a comprehensive project providing secure accesses to clinical data via WWW, namely PCASSO (Patient-Centered Access to Secure System Online) [19, 33, 124, 125]. PCASSO is a research, development, and evaluation project to exploit state-of-the-art security and WWW technology for health care, intended to provide secure access to clinical data for health care providers and their patients using Internet. In what follows, we in turn examine security measures taken at the server side, the client side, and the internet link between server and client.

1. **PCASSO Server**

PCASSO addressed server vulnerabilities by hosting its server and clinical data repository on a high-assurance OS (Data General's B2 DG/UX). Of the rating defined in the Department of Defense Trusted Computer System Evaluation Criteria, Class $B2$ provides PCASSO with the following security functions: user authentication, identity-based (discretionary) access control, label-based (mandatory) access control, finely grained privileges, auditing, and trusted communication path. DG/UX's advanced label-based access control mechanism protects PCASSO executables from virus infection, and patient data from access by unauthorized software. The server uses host and packet filters that prohibit administrative access from any machine other than trusted local machines. Advanced logging capabilities monitor critical aspects of PCASSO execution, and administrative tools allow the system administrator to query and analyze the audit trail to review system behavior, to identify potential system misuse and intrusion, and to view statistical reports. The principal architectural components of the PCASSO Server are the Administrator, the Importer, the Encryption service, the RDBMS, the hypertext transfer protocol daemon (HTTPD), and the File System.

Administrator. The Administrator is a trusted application that provides the PCASSO system administration capabilities.

Importer. The Importer is a trusted application responsible for importing patient record information from the UCSD interface engine (in HL7 format) into the PCASSO Clinical Data Repository (CDR) and Research Data Repository (RDR). The Importer is also responsible for establishing the initial sensitivity labels for all information.

Encryption Service. The Encryption service provides an end-to-end confidential, authenticated communication service for the PCASSO Client-Server Protocol (PCSP), which is based on SSL.

RDBMS. The RDBMS is a Class $B1$ Oracle relational database management system (Trusted Oracle 7.2) that is responsible for managing the patient-centric Clinical Data Repository (CDR), which contains the patient records, and the research data repository (RDR), which contains only non-patient-identifiable information and has a non-patient-centric organization. The CDR and RDR are separate Oracle instances. These databases physically reside within operating system files dedicated to the Database Domain. Stored procedures managed by the RDBMS log accesses to the CDR, and store these data within patient records, enabling access through the normal patient-record interface.

HTTPD. The HTTPD is a standard Internet Web server that provides the initial login Web page to allow access to the PCASSO System from Internet Web browsers via the HTTP protocol. File System. The DG/UX file system contains all the files that store information within the PCASSO System.

2. PCASSO Client

PCASSO clients are normally PCs running windows 95. Because Windows95 lacks architectural features fundamental to security (e.g., self-protection, process/virtual address space isolation), they must be considered high-risk from an assurance perspective. The PCASSO client software provides features and countermeasures in its design and implementation to raise the level of sophistication and costs necessary to compromise patient data. They include the following:

(1) The client is designed as a read-only system, providing no mechanisms for the display applets to store sensitive patient data to the client computer's file system. In addition, the client's Web browser does not cache HTML, images, or Java code when interacting with sites through secure channels, thus reducing the risk posed by malicious

applications or users who browse or poison secondary storage devices for cached patient information or applet code.

(2) The client is provided encryption services that implement authentication and end-to-end confidentiality. These encryption services prevent lower-layer malicious network protocols or device drivers from eavesdropping on data as they are exchanged between the Java Virtual Machine and the network interface.

(3) Client applets are stored on the PCASSO server and downloaded upon the establishment of the secure connection. By avoiding storage of client applets on the insecure Windows file system, PCASSO eliminates the risk that binaries may be modified or replaced.

(4) Client applets are restricted to the Java runtime "sandbox", thus providing a boundary of containment as to where the data handling and display applets can store or forward sensitive patient records.

(5) Client applets check for other applets execution, thus reducing the potential of interference or subversion.

(6) The client stores no authentication data on the Windows 95 machine; these data are stored on a read-only diskette that is inserted only when the client user is instructed to do so, thus reducing the window of opportunity available to entities attempting to compromise the client's authentication information.

(7) Client applets avoid the use of keyboard entry for inputting user authentication data; instead, Java widgets employ graphical mouse-oriented interfaces that allow users to enter sensitive information, while limiting the exposure of these data to capture (e.g., keyboard interrupt monitoring). The server maintains the association between

the connection and the PCASSO User ID, allowing the client to clear the ID, password, and key password, from memory to avoid capture by a malicious application.

(8) Applets convert textual information to images to increase the difficulty for malicious processes to search resident data structures for sensitive information.

3. Internet Link

All the data in transit are encrypted by PCASSO Client-Server Protocol (PCSP) that is based on SSL.

With all these measures in place, PCASSO provides assurance of correct operation through formal, disciplined design and development methodologies, as well as through functional and penetration testing. It is evident that PCASSO implemented a secure end-to-end health care system, with emphasis on keeping health data secure from outside attackers. Our study in this dissertation goes beyond building a secure end-to-end system for achieving data security against outside attackers, and we focus on addressing individual privacy, which further involves data protection towards legitimate parties.

2.2 Access Control in Health Care

Access control constitutes a fundamental part of data security within organizations, and has been an extensively studied subject in literature. We believe many existing studies are instrumental in developing a practical access control mechanism for health care systems. In this subsection we give a brief survey on access control and mainly RBAC (role based access control) in health care, and list features that are useful and should be implemented in RBAC for health care applications. These features are a good reference and starting point for developing a practical access control mechanism for health care systems.

R. Anderson [11] proposed a security policy model for health care information systems that allows the British Medical Association (BMA) to meet security requirements of the electronic patient record (EPR) and is the base of any proposed system claims to operate the EPR. Anderson's model is composed of a set of principles based on a statement found in the Good Medical Practice booklet issued by the General Medical Council (GMC), which says: *patients have a right to expect that you will not pass on any personal information, which you learn in the course of your professional duties, unless they agree.* This model is the first security policy model that spells out clear and concise access rules for clinical information system. Aljareh and Rossiter [9] gave an in-depth analysis of Anderson's model including the difficulties for practical implementation.

In the last few years, role based access control (RBAC) and its variants have acquired a great importance among access control models. Traditional discretionary access control (DAC) bases authorization decision on discretion of individual users, thereby inapplicable to the majority of health information; the most commonly used mandatory access control (MAC) in the form of multi-level mechanism that associates information with labels as “TOP SECRET”, “SECRET”, and “CONFIDENTIAL” is not sufficiently flexible for industry use [26]. In contrast, in RBAC, rights and permissions are assigned to roles rather than to individual users; users acquire these rights and permissions as they activate appropriate roles. Such an idea greatly eases the administration of authorization, as roles are stable while users are volatile in an organization. The major benefits of RBAC are the ability to express and enforce enterprise-specific security policies and to simplify the process of security management.

It was accepted [26] that RBAC is a more suitable access control model than other models for health care systems, since health care practice tends to manage access rights to patient data around medical “roles” that associate with individual patients. Numerous effort has been dedicated to investigating application of RBAC and its variants to health care setting. [102] presented a demonstration of the use of RBAC with patient records, proving usefulness and applicability of RBAC to health care systems. [80] reported a preliminary framework to integrate hierarchical access control into the health care system. In particular, it identifies four kinds of principals, i.e., clinically-qualified staff (e.g., physicians, nurses), non-clinically qualified medical staff (e.g., technicians, secretaries), non-medical staff (e.g., programmers), patients; furthermore, a document is partitioned into Q ranked equivalence classes with the property that an authority with permission P_m can decrypt all parts of the document encrypted at equivalence classes E_j provided $m > j$. Consequently, master key holders can decrypt the entire document, while more restricted users can only decrypt those parts of the document for which they hold the appropriate key authority. Main weaknesses associating with this framework include coarse granularity of the principal classification and involving no contextual information.

As a high-level access control model that uses the abstraction concept of role to reduce the complexity of authorization management, RBAC requires intermediate structures to implement its abstraction concepts on lower level access control on a platform. Domain and Type Enforcement (DTE) model [28], a lower level mandatory access control mechanism, can be used to predate RBAC towards that end. In DTE, subjects (or transaction programs) are assigned “Domain” labels and objects are assigned “Type” labels. Associating with each Domain-Type pair is a set of allowable access modes. The data structure that contains access modes for all Domain-Type pairs is called the Domain-Type Access Matrix. The system in [54] used a combination of RBAC and DTE, augmented with a logic-driven authorization engine, in an attempt to construct a dynamic authorization framework supporting multiple authorization types in health care systems. Contextual constraints are implemented in this framework, such that each individual authorization request is assigned a type and the conditions needed to satisfy the requirements for that authorization type are checked dynamically. Once the conditions are checked using certain context information, the DTE subject-domain table is read to assign the correct domain (based on the invoked subject) to the user session. The actual permissions required for the subject to carry on its intended operation are read off from another DTE table - i.e., the Domain-Type Access Matrix.

eMEDAC [130, 131] enhanced an original MEDAC policy that was a three-layer model with role based concept being introduced but not fully implemented. eMEDAC instead reinforced the role-based concept to better embrace MAC and DAC. eMEDAC exploited the concept of a Hyper Node Hierarchy (HNH) to inherit permissions (discretional control) and derive security labels (mandatory control) instead of retrieve them as stored static labels from database. This gave a more flexible access control and save of storage space. HNH are used to construct User Role Hierarchies (URH) and Data Set Hierarchies (DSH) and to derive the security labels (consisting of a security level and a category set) of user roles and data sets. Flexibility however comes at expense of efficiency. [77] further improved eMEDAC by combining team based concept with role based concept. A primary contribution of such a combination is the feasibility to accommodate in a natural way the context information in the role based access control. From this perspective, this method is a well founded context based access control model, as "Team" includes with itself context information consisting of user context and object context. This model is able to, on the one hand, leverage the scalable security administration benefits of RBAC, while one the other provide fine-grained permission activation and deactivation to individual users and object instances. For example, it is thus possible to assign and administer broad permissions for doctors on object types based on some role definitions and yet activate a doctor's permission to a patient's records (object instances) only when he is taking care of the patient. An implementation of this model in a relational database management system for a health care organization was given. In [129], this role-based policy was again implemented in a decentralized manner by virtue of digital certificates for authentication. In particular, three types of digital certificates are used: Identity Certificate (IC) for authentication; Attribute Certificate (AC) for authorization; and Access-Rule Certificate (RC) for propagation of access control policy. Compared to an IC, an AC contains attributes that specify access control information associated with the AC holder (such as group membership, role, security clearance), and normally it has comparatively shorter lifetime without revocation mechanisms. An RC is a long-lived certificate with revocation mechanisms, containing digitally signed sets of rules. RCs enable parties responsible for policy to create and distribute access control mechanisms remotely and securely and to create rules authorizing access to their respective resources. Jurecic and Bunz [99] also described a prototype implementation of role-based access control in combination with attribute certificates. In this implementation, however, attribute certificates along with other security related public keys, are stored in an Organizational Directory dictated under X.500.

Tzelepi and Pangalos [178] presented an extended RBAC model where permissions for access are given based on the semantic content of images. Two extensions are done to regular RBAC: one introduces the user attribute base which is defined as ⟨user_id, user_name, domain, location⟩; the other extends the general form of Role-Permission relationship as ⟨identifier, s: role, action, t: target, constraint(s, t)⟩, where s is defined as a role and t is defined as an object on which actions can be performed. Access right is granted only when the constraint(s, t) is satisfied. Essentially, these extensions attempted to accommodate context information involved in a request for access to certain patients' medical information. Task-based access control model (TBAC) [90] can be a good complement to RBAC in health care systems. TBAC implements *purpose binding* such that a user is granted access only if such an access is necessary for the user to perform her/his current task and if she/he is authorized to perform this task. RBAC is not designed to directly enforce purpose binding.

Access control in health care must at minimum meet the stipulations of laws, legal regulations. With HIPAA [86] taking effect in U.S., many work explored developing access control systems in compliance with HIPAA requirements. For example, [166] illustrated an example of implementing context-based access control into health care setting. Simply speaking, context-based access control is built over either user-based or role-based access control, but going a step further: access control decisions in user-based or role-based access control answer questions for example, "Should this person (or a person who performs this job function) be allowed to access this type of data?"

while the equivalent context-based question would be, “Should this person (or a person who performs this job function) be allowed to access this type of data as it applies to this particular patient?”. The discussion in [166] made use of stored procedures in a relational database to check some context information constituting a to-be-satisfied condition prior to bestowing the access control privileges; moreover, various aspects concerning a successful implementation of the model such as the dependency of the model on core operating system and database security controls were emphasized. Cole [50] discussed the applicability of RBAC to achieve HIPAA compliance in health care systems, and provided many useful suggestions on practical implementation.

Other access control models were also studies for health care systems. For example, [66] explored the possibility of applying Partition Rule Based Access Control (PRBAC) to civilian uses such as in health care practice. PRBAC is an advanced computer access control technology for computer network applications. PRBAC is adaptable to the needs of specific communities, allowing the authorities responsible for the security policy of a community (partition) to define the rules for controlling access to sensitive information within the community, enforce those rules within their community, as well as to relate their rules with those of another community to enable information to be exchanged in a controlled fashion. In essence, PRBAC involves conveying authorizations in X.509 type certificates, target data sensitivities in labels, and security policies in PRBAC Information Files (PIFs). PRBAC provides a standardized Access Control Decision function that compares user authorizations against target data labels according to a defined security policy. The ability to specify widely varying security policies in PIFs provides the PRBAC mechanism great flexibility to meet the needs of users as diverse as in health care systems.

From what we have reviewed so far, it is clear that RBAC is a right access control model for health care systems. But RBAC by itself may not be enough in some circumstances, and it has to be extended in many aspects to meet the dynamic, complex setting and varying security demands in health care systems, or for satisfying legal requirements. To that end, in what follows we emphasize many features that should be incorporated in RBAC. Developing such an access control module in practice may require considerable engineering effort, but implementation of some of the features may follow a modularized approach, providing "features on demand".

- While it supports authorization at the discretion of roles in general, the access control module should also supports user-based access control. User-based access control is a useful complement to RBAC in many circumstances, and is further a requirement by HIPAA. Implementing user-based access control in RBAC can refer to [149, 185]. Augmenting RBAC with mandatory access control property also has advantages in health care systems [54, 130].
- Due to the discrepancies among health care organizations, there may be differences between *organizational roles* and *system roles*. An organizational role is a natural position in an organization, while a system role refers to a capsulation of access rights in the target system. Separation of organization roles and system roles is effective in system design and maintenance for health care systems. The methods suggested in [142] should be incorporated into RBAC as basic elements.
- Access rights to heath data may change dynamically, depending on the context. Various contextual factors, such as purpose of requests (purpose binding), affil-

iation of the requester, location and time of the requests, relationship between the requester and the patient whose data are being requested, should be considered. The contextual factors can be implemented as a set of constraint logics commanding RBAC. The reference of [54] is a good starting point.

- Access control in health care systems involves protection of *individualized* data. *Informed consent* must be implemented such that health care organizations cannot use health data for purposes other than consented to by patients. Implementation of informed consent is a practice for compliance with HIPAA. [42] discussed several issues associating with implementing informed consent. At the technical level, a way to implement informed consent can follow the method in [2], where a construction of a centralized, anonymous medical record repository was presented.
- Finer access granularity with respect to different data modalities should be achieved. Taking medical images for example, access restriction can be enforced at the block level and not simply at the file level, semantic content of an image should be extracted and specified. [177] presented such an access control mechanism over medical images.
- Routine transactions in health care systems normally involve multiple entities within individual organizations or even across organizational boundaries. Moreover, joint projects among many organizations such as health care provider and medical research institutes are also common. As a result, access control in health care systems should possess workflow support. Access control with workflow support must implement strict separation of duty and principle of least privilege. [64] extended and adapted RBAC to provide adaptive authorization for workflows.

- Access control for health care system must accommodate exceptional accessing, while without compromising the overall system security: emergency access and legal auditing are two exemplary exceptional accesses of health data. Exceptional accessing is either essential due to the nature of health care systems, or for fulfilling law requirements. [2] gave a solution to emergency access of health data by virtue of emergency access tokens.
- RBAC adheres to the principle of *general denial with explicit consent*, but authorization taking the form of *general consent qualified by explicit denial* is of particular use in some situations. For example, it is often required to enforce authorization policies similar to: access is granted to Physicians except for Dr. John, who is the patient's father in law. Augmenting RBAC with such explicit denial of authorization offers additional flexibility in access control. [149] presented a solution to this problem.
- Support of delegation of access rights and responsibilities offers another level of flexibility in authorization. Two types of delegation should be specified in health care systems: doctors can delegate part of their privileges to other qualified personnel; patients can delegate their control on the health information to trusted guardians in case of emergency. [190] proposed a delegation framework based upon RBAC.
- Strict auditing must be implemented. All accesses to health data must be documented non-repudiatable.

These are typical features we believe to be useful for access control in health care systems. Depending on particular applications, more features may be needed.

CHAPTER 3

Building A Unified Trust Infrastructure for Health Care Organizations

In this chapter, we discuss establishing a unified trust infrastructure for health care systems, and more precisely we focus on user authentication in health care setting. We must point out that a trust infrastructure involves a broader range of issues other than user authentication, e.g., policies, models, etc, but discussion of them is out of the scope of this dissertation. User authentication is a fundamental and enabling service for organizational and inter-organizational trust establishment. Discussions in this chapter thus lay a foundation to solve other security and privacy issues in later chapters and beyond this dissertation.

Health care systems represent a complex and cooperative setting as signified by Figure 1.1. Due to the varying operational environments and contexts, systems, and security requirements, many authentication techniques are necessary for their respective advantages. For example, within an organization, it is of particular convenience for interior physicians and administrative staffs to be authenticated for their routine work by using passwords; in contrary, for inter-organizational cooperation, user authentication relying on credentials demonstrates advantages because of higher reliability and expressiveness.

It is thus of interest and importance to investigate tailoring multiple user authentication techniques into a unified infrastructure that can provide flexible authentication services to health care organizations.

3.1 Tailoring User Authentication Techniques Towards A Unified Trust Infrastructure

User authentication is an enabling service for trust establishment within individual organizations and across organizational boundaries. Through trust establishment, users prove their qualification and legitimacy for certain services or data. Technically, trust establishment is accomplished by virtue of user authentication (or identification) techniques. A variety of user authentication techniques and modes have been studied and used in practice, e.g., password, identity certificate (e.g., X.509 digital certificate [186]), attribute-based certificate, anonymous credential, group signature, and so on[1]. We next give a short introduction on these techniques and their potential uses in health care systems.

Password: Entry of a user ID followed by a password is the most commonly used means of user authentication since the advent of computers and is still gaining popularity especially among mobile users. In a password authentication system, each user shares a password or some simple password verification data (PVD) derived from the password with a single server, and the user only needs to memorize the password and uses it in

[1] Note that biometrics (e.g., [25]), tokens such as smart card and SecurID card are also useful authentication means in health care systems. But they are normally used for physical access, not for negotiating trust among several parties, which is our concern.

user authentication process. One apparent advantage of using password lies in that it has little or no actual cost since no associated physical accessories such as smart cards, sensors, scanners are required. Password authentication still demonstrates vitality in health care systems. For example, it will be of great convenience for physicians to authenticate to their routine services from within care organizations. Moreover, in some places such as outpatient clinics where dumb terminals are still used, passwords may be the only option for user authentication.

However, because of the limited dictionary space where passwords are drawn, password authentication is susceptible to brute-force dictionary attack, whereby an attacker enumerates every possible password from the password dictionary for either repeated one-line logins or off-line checks against a valid login transcript. The former, known as on-line dictionary attack, can be thwarted at the system level by limiting the number of unsuccessful login attempts made by a user. In contrast, the latter, known as off-line dictionary attack, is harder to resist. The bulk of research has been dedicated to the development of password (only) authentication systems that are robust against off-line dictionary attack by outside attackers, e.g., [20, 31, 112]. In practice, attackers take on a variety of forms, such as disgruntled system administrators, hackers, viruses, worms, accidents and mis-configurations. As a result, no security measures and precautions can guarantee that a system will never be penetrated. Once the server housing user passwords or PVDs is compromised or corrupt, all passwords or PVDs may fall in the hands of bad guys. Hence, concern of off-line dictionary attack against the server database arises. Existing solutions to this problem are distributing PVDs and the authentication functionality to multiple servers, e.g., [69, 134, 150]. Unfortunately, these methods may be hard for practical use since a user has to simultaneously contact multiple servers for user authentication. Clearly, an ideal multi-server password authentication system should allow users to communicate with only one server as in a single-server system while the authentication data and functionality are still distributed. Shortly, we present a two-server system that on the one hand solves the limitations of the above multi-server methods, while on the other hand possesses compatibility with other user authentication techniques.

Identity certificate: Public key cryptosystems (digital signature and public key encryption) are absolutely necessary for trust establishment in open, distributed and cooperative systems such as health care, where it is hard to pre-establish or maintain secrets shared among different parties. A public key cryptosystem uses two different keys: one is a private key and the other is a corresponding public key; the private key is for creating digital signatures or decrypting the ciphertexts, and the public key is for verifying digital signatures or creating ciphertexts. A crucial step associating with the use of public key cryptosystems is binding public keys with their key holders. This is achieved through *certification*, a mechanism enables a ready assertion of the association between a public key with its owner. The standard certification framework is X.509 Public Key Infrastructure (PKI) [186], where a (or more) trusted third party, known as certification authority (CA), issues digital certificates on users' public keys. X.509 certificates certify public keys at the discretion of globally unique individuals, thereby in nature identity certificates. In a PKI, as long as multiple CAs are involved, they are normally organized into a tree hierarchy and at least the root CA is universally trusted.

Attribute-based certificate: As discussed earlier, a X.509 certificate is essentially an identity certificate, binding a public key to a globally distinguished name. Sometimes, a user's name is not essential for authorization decisions. What really matters is whether the key holder in question possesses a certain qualification, property or has been authorized for certain access rights. This leads to the concept of attributed-based certificate such as SPKI (Simple Public Key Infrastructure) [65]. Naming conventions represent the main differences between attribute-based certificates and X.509 certificates: the name in an attribute-based certificate needs not necessarily to be globally unique and the certificate tends to asserts that the key holder possesses some qualification or attributes. Based on the basic concept of attribute-based certificate, many variants can be constructed, e.g., role certificate. Role certificate will provide a direct support for RBAC in health care systems. Moreover, attribute-based certificates are better to be implemented as short-term credentials, which have wide applicability in inter-organizational transactions due to the advantages on key management.

Anonymous credential: Anonymous credential, first introduced in [45], is an enabling technique to achieve strong user privacy in user authentication. Upon *regular* credentials such as identity certificate and attribute-based certificate, anonymous credentials enable user authentication to proceed in an anonymous manner, that is, different authentication executions by the same user cannot be linked (unlinkability). While regular credentials can be adapted to be *pseudonymous*, they are unlikely to achieve unlinkability. With individual privacy becoming increasingly a concern, anonymous credentials are now more and more emphasized and should be included as an important privacy enhancing technique in health care systems. This however requires serious considerations to accord with the deployed organizational access control mechanism that normally requires recognizing and identifying users. The idemix system [48] implemented state-of-the-art techniques for anonymous credential and demonstrates many desired features.

Group Signature: Group signature is another commonly used privacy enhancing technique, first introduced in [47]. Informally, a group signature scheme involves a group of users, each holding a distinct group signing key; using the signing key, a user can issue a publicly verifiable signature on behalf of the group; given a group signature, one cannot determine the actual signer, nor can one determine whether two signatures were due to the same signer; in the event of dispute, the *group manager* (GM), holding a secret revocation key, can "open" the signatures and reveal the real identity of the signers. The intuition behind group signature as a user authentication technique is that the ability to issue valid groups signatures asserts a user's membership to a certain group. This may be useful in many scenarios in health care systems. As with anonymous credential, health care systems should accommodate group signature in their authentication infrastructure. The group signature scheme proposed in [1] represents the current state of the art.

These are some typical user authentication techniques and modes that are useful to health care systems. While these technique can be considered and deployed independently within individual organizations, it is clearly of huge interest to unify them into a coherent infrastructure providing flexible authentication services to health care organizations. Let us first see an observation: in health care systems represented by Figure 1.1, an organization needs to verify not only its affiliated users, but also users from other organizations such as partners and business associates. As such, the authentication service within an organization must support both organizational and inter-organizational trust establishment. Authentication techniques depending on public key cryptosystems, e.g., identity certificate and attribute certificate, are apparently indispensable in such settings where pre-establishing of secrets is often not feasible and practical. We therefore suggest each organization establish a *security manager* dedicated to certification, and other security related matters (e.g.,handling individual privacy, and issuing secret keys to the affiliated users whenever necessary). Given the security manager is established, the fundamental principle for inter-organizational authentication would in general follow a two-level authentication procedure as outlined in Figure 3.1: (1) a user first authenticates to the organization she belongs to, who in turn certifies her, e.g., issuing her a credential. Note that this *organizational* authentication procedure is not necessarily on-line with respect to the following inter-organizational access; (2) then when the user initiates an inter-organizational access, the authenticating organization verifies the authenticity of the certification. It is clear that each organization essentially establishes organizational PKI, with the security manager acting as the CA (certification authority). Depending on applications, intermediate CAs may be allowed and included in

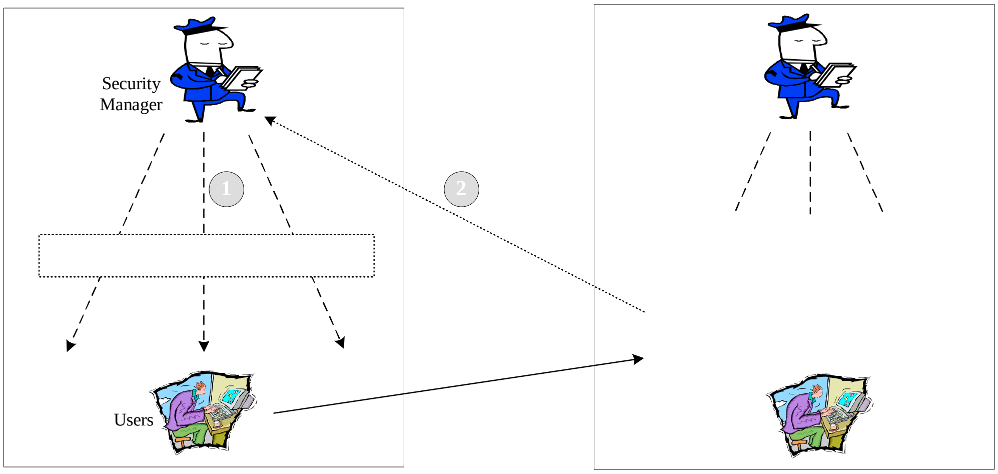


Figure 3.1: Two-level Inter-organizational Authentication Procedure.

the certification path within an organization, in which case, the security manger would serve as the root CA. For example, each clinical unit or department in a hospital certifies its own users while the organizational security manager certifies the clinical units and departments. In practice, the establishment of organizational PKIs may depend on external PKIs such that security managers themselves are certified by an outside CA. Whatsoever, trust between organizations is eventually established by and traced back to the respective organizational security managers. It is important to note that the security manager is not necessarily a single entity. Instead, for efficiency and security reasons, a security manager may be a group of entities, working for example in a threshold manner for a single purpose, or for different purposes. The exact form of security manager depends on particular organizations or applications.

We must stress that the establishment of security manager is a practice compliant with HIPAA [86] (section 164.512(f)), which recommends to set up "privacy officers" for overseeing compliance with privacy regulations and policies. Moreover, security manager acting as a trusted third party (TTP) is of absolute necessity in many security systems. For example, in a system achieving *revokable* user anonymity, enlisting a TTP may be the only way to avoid complete user anonymity.

It is now evident that a way for establishing an organizational trust infrastructure is to unify the above various user authentication techniques around the organizational PKI with the security manager acting as a CA or TTP, as outlined in Figure 3.2. It is noted that the architecture is an open system in the sense that new authentication techniques can be continuously incorporated into the infrastructure for emerging needs. It is easy to see that except password, the organizational PKI readily accommodates all

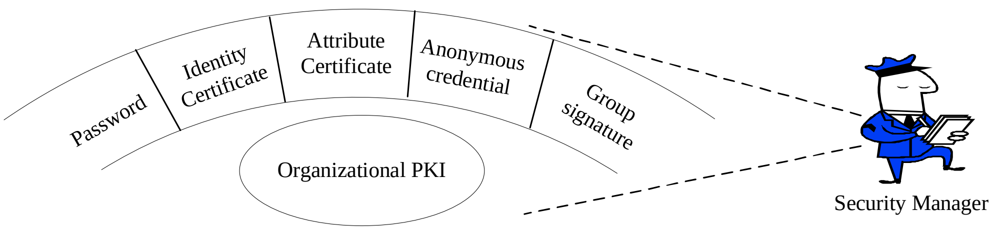


Figure 3.2: A Unified Organizational Trust Infrastructure.

other authentication modes. For example, in identity certificate and attribute certificate, the security manager is a CA issuing certificates; in an anonymous credential system, the security manager is an issuing organization that issues anonymous credentials to users [45]; in a group signature system, the security manger is the group manager (GM) responsible for system setup, group membership management and anonymity open [1, 47]. What clearly remains is to integrate password authentication into the organizational trust infrastructure. To that end, we propose a novel two-server password authentication system that exploits the security manager for managing a back-end server (the other server is naturally a service server operated by the corresponding service provider in the care organization). Another motivation for this two-server password system is to eliminate the single point of vulnerability inherent in traditional single-server systems. As a result, concern of off-line dictionary attack initiated at the server side, e.g., in the event of attacks by unscrupulous insiders or break-ins by outside attackers, is resolved.

3.2 A Two-server Password Authentication System

We first introduce a two-server architecture. Then we present our authentication and key exchange protocol using password[2] upon the proposed two-server architecture: for ease of security analysis, we first give a preliminary protocol, based on which we then develop our final protocol by circumventing the weaknesses in the preliminary protocol.

3.2.1 A Two-server Architecture

The two-server architecture we propose is shown in Figure 3.3: There are two servers on the server side, a *service server* and a *central server*. The front-end service server is the actual one providing a certain service to users and hence is configured to be contactable by users. The central server stays back-end and thus transparent to the public, and its objective is to assist the service server in user authentication and key exchange. The central server is managed by the organizational security manager. This architecture allows us to distribute user passwords and the verification functionality to the two servers in order to eliminate the single point of vulnerability in the traditional single-server model. This positioning of servers has two salient advantages: (1) since subject to no direct exposure to the public, the central server is less likely to be attacked, which in turn increases the overall security of the architecture; (2) users only need to communication with the service server, hence the demand of bandwidth as well as synchronization at the user side is substantially decreased.

In practice, the central server is configured to support multiple service servers, each

[2] It is common in a password system that a user not only authenticates to but also negotiates a session key with the server for subsequent data exchange.

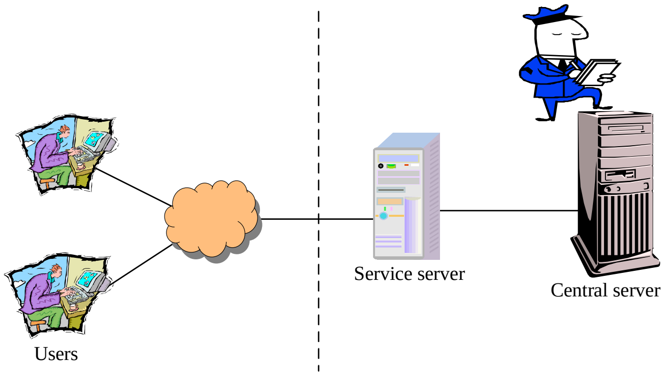


Figure 3.3: A Two-server Architecture.

providing a service and managed by a different unit or department within a health care organization, as illustrated in Figure 3.4.

3.2.2 A Preliminary Authentication and Key Exchange Protocol Using Password

The basic idea of our protocol for the two-server architecture is to distribute the password verification data (PVD) and verification functionality to the two servers, for the purpose of eliminating the single point of vulnerability. In particular, a user's short password is split into two long secrets and each is hosted by a server, and neither of the servers can compromise user passwords by means of off-line dictionary attacks. In other words, without compromising both servers, an attacker cannot recover users passwords by dictionary attacks. We start by listing the notations that are used in the sequel in this chapter, for ease of referencing.

High level description: Three types of entities are involved in the system, i.e., users, service servers and a central server. Users only see the service servers, and the central server (CS) is hidden from the public. A service server (SS) acts as the relaying

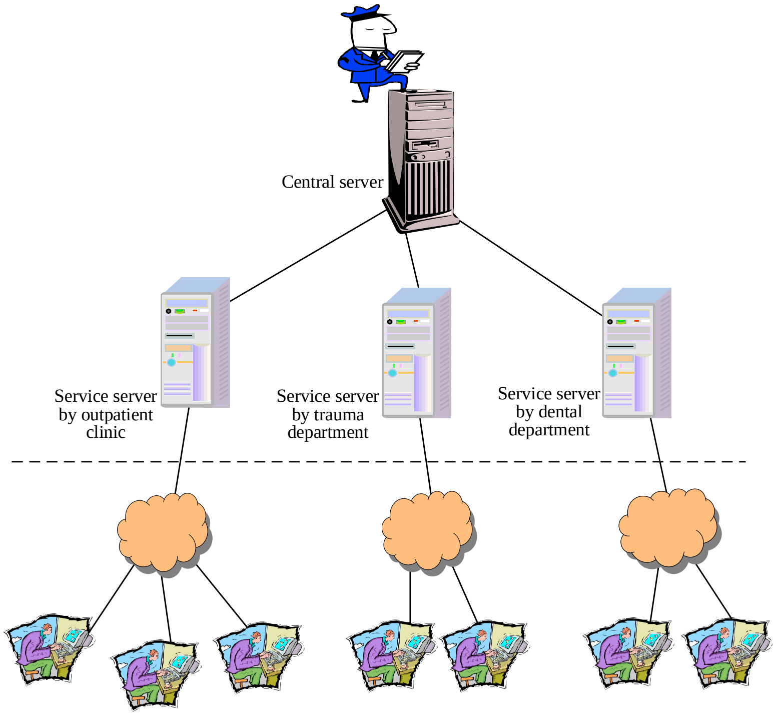

Central server
Service server by outpatient clinic
Service server by trauma department
Service server by dental department

Figure 3.4: Central Server Supporting Multiple Service Servers.

party between its users and CS. In this setting, an important observation is that CS clearly assumes more trust than a SS because of sufficient expertise and funds that the security manager has, together with the fact that CS does not directly expose to the public. Considering such asymmetry in terms of trust upon CS and SS, adversary model in our protocol is that *CS is semi-honest and each SS is malicious, and CS does not collude with any SS*. More specifically, CS is honest-but-curious [78], i.e., it follows the protocol specification, with the exception that it may try to derive extra information by analyzing the protocol transcript (CS is even allowed to eavesdrop on the communication channel between U and S); on the contrary, a SS may act arbitrarily for uncovering user passwords. Moreover, in this preliminary protocol, we assume a secret communication channel between a SS and CS, which can be established by the two parties sharing a

P, q, p	three large primes such that $P = 2p + 1$ and $p = 2q + 1$.
g_1, g_2	elements in QR_p, where QR_p represents the group of quadratic residues modulo p. The discrete logs to each other is not known.
g_3	an element in QR_P.
π	a user's password.
$h(.)$	a cryptographic hash function modelled as a random oracle [38].
U, SS, CS	identity of a user, a service sever and the central server, respectively.

Table 3.1: Notations for Two-Server Password System

secret key.

Each user U has a short password π, and π is transformed into two long secrets π_1 and π_2, each of which is registered to the service server SS the user belongs to and the central server CS, respectively, in a out-of-band *registration* phase. During *authentication*, SS and CS together produce a challenge to U using their respective π_1 and π_2. U responds by applying his password π. With the responding data, SS and CS help each other verify the authenticity of U. Upon the servers validating U, they reply to help U authenticate them. In the meantime, SS and U establish a common session key for subsequent data exchanges.

User registration: in any password system, to enrol as a legitimate user in a service, a user must beforehand register to the service provider by establishing a password with the provider. In our system, a user U needs to register to not only the actual service provider SS but also the central server CS. Suppose U has already successfully identified to SS, e.g., by showing his identity card, U picks and splits his password π into two long number $\pi_1 \in_R Z_q^*$ and $\pi_2 \in_R Z_q^*$ such that $\pi_1 + \pi_2 = \pi \pmod q$, where q is a prime as defined in Table 3.1. U then registers in a secure way π_1 and π_2 to SS and

CS, respectively. As a result, SS stores the account information (U, π_1) to its secret database, and CS stores (U, π_2) to its secret database. One may wonder how U registers π_2 to CS, as CS is supposed hidden from U. This is not a problem in practice: U can contact the security manger that manages CS by normal mail, etc. Upon completion of the registration, U can request service from SS, by exploiting the protocol in Figure 3.5 for authentication and establishment of a common session key.

The protocol: let p, q, g_1, g_2 and $h(.)$ be defined as in Table 3.1; we have omitted modulo p for arithmetic operations in Figure 3.5, as this is clear from the context; we also omitted a session ID *SID* for each message and *SID* serves to prevent replay attack. We next follow the protocol step by step. To initiate a request for service, U sends his identity together with a service request *Req* to SS in $M1$. SS first relays the request

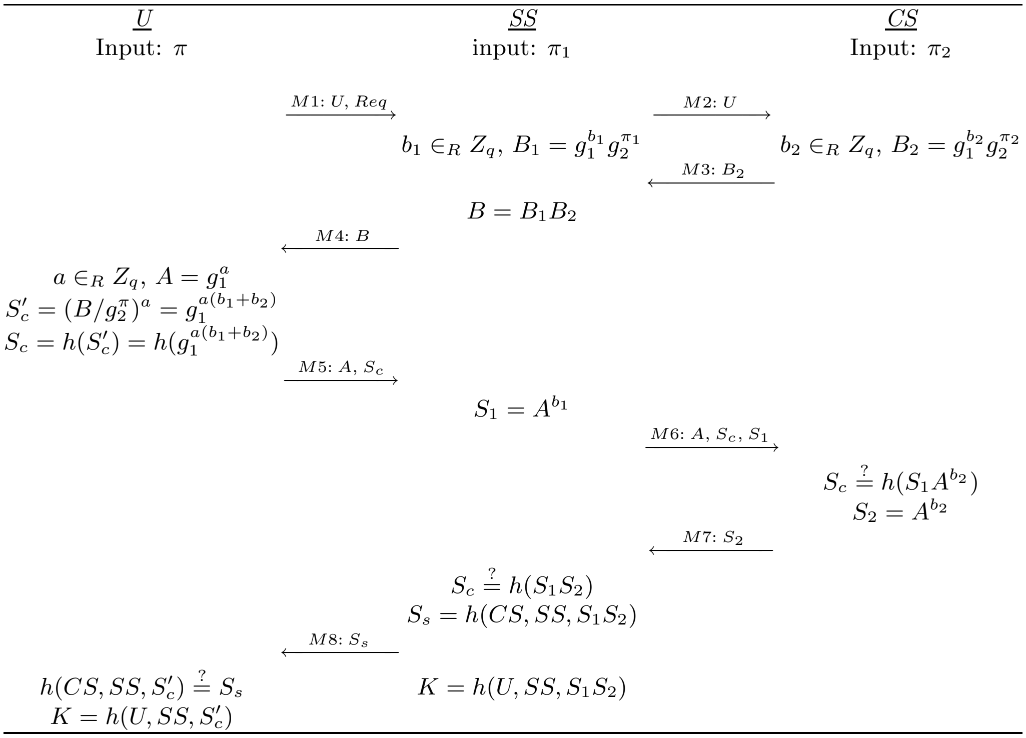

$\underline{U}$
Input: π
$\underline{SS}$
input: π_1
$\underline{CS}$
Input: π_2
$\xrightarrow{M1:\, U,\, Req}$
$\xrightarrow{M2:\, U}$
$b_1 \in_R Z_q,\ B_1 = g_1^{b_1} g_2^{\pi_1}$
$b_2 \in_R Z_q,\ B_2 = g_1^{b_2} g_2^{\pi_2}$
$\xleftarrow{M3:\, B_2}$
$B = B_1 B_2$
$\xleftarrow{M4:\, B}$
$a \in_R Z_q,\ A = g_1^a$
$S_c' = (B/g_2^{\pi})^a = g_1^{a(b_1+b_2)}$
$S_c = h(S_c') = h(g_1^{a(b_1+b_2)})$
$\xrightarrow{M5:\, A,\, S_c}$
$S_1 = A^{b_1}$
$\xrightarrow{M6:\, A,\, S_c,\, S_1}$
$S_c \overset{?}{=} h(S_1 A^{b_2})$
$S_2 = A^{b_2}$
$\xleftarrow{M7:\, S_2}$
$S_c \overset{?}{=} h(S_1 S_2)$
$S_s = h(CS, SS, S_1 S_2)$
$\xleftarrow{M8:\, S_s}$
$h(CS, SS, S_c') \overset{?}{=} S_s$
$K = h(U, SS, S_c')$
$K = h(U, SS, S_1 S_2)$

Figure 3.5: A Preliminary Authentication and Key Exchange Protocol Using Password.

to CS by sending the user ID in $M2$, and then selects a random number $b_1 \in_R Z_q$ and computes $B_1 = g_1^{b_1} g_2^{\pi_1} \pmod{p}$ using his password share π_1. Upon receiving $M2$, CS chooses a random number $b_2 \in_R Z_q$ and computes $B_2 = g_1^{b_2} g_2^{\pi_2} \pmod{p}$ using his password share π_2. CS then sends B_2 in $M3$ to SS. Upon reception of B_2, SS computes and sends $B = B_1 B_2 \pmod{p}$ to U in $M4$. After receiving $M4$, U selects $a \in_R Z_q$, and computes $A = g_1^a \pmod{p}$, $S'_c = (B/g_2^{\pi})^a = g_1^{a(b_1+b_2)} \pmod{p}$ and $S_c = h(S'_c)$, respectively. U then sends A and S_c to SS in $M5$. Getting the message, SS computes $S_1 = A^{b_1} \pmod{p}$ and sends A, S_c and S_1 to CS in $M6$. Upon reception of $M6$, CS computers $S_2 = A^{b_2} \pmod{p}$ and checks whether $S_c \overset{?}{=} h(S_1 S_2) = h(g_1^{a(b_1+b_2)})$: if it holds, CS is assured of the authenticity of U, and continues the protocol by sending S_2 to SS in $M7$; otherwise, CS aborts the protocol.

Assuming SS receives S_2 in $M7$, it checks whether $S_c \overset{?}{=} h(S_1 S_2)$: if it holds, SS is convinced of the authenticity of U. At this stage, both servers have authenticated U. SS then computes and sends $S_s = h(CS, SS, S_1 S_2)$ to U in M_8. Immediately after that SS computes a session key $K = h(U, SS, S_1 S_2)$ and grants U with the requested service over a secure channel protected using the session key K; otherwise, SS rejects and aborts the protocol. Upon reception of M_8, U checks if $h(CS, SS, S'_c) \overset{?}{=} S_s$: if it holds, U has validated the servers and then computes a session key $K = h(U, SS, S'_c)$.

Remarks: In order to prevent leakage of even one bit of information in π, π_1 and π_2, respectively, ideally the corresponding entities should choose their respective random numbers as even numbers.

Security discussion: in what follows, we analyze the security of the above protocol. Our analysis is based mainly on the following well-known Decisional Deffie-Hellman

(DDH) assumption [21]:

> *DDH Assumption*: let p, q be defined as in Table 3.1, and $g, h \in_R QR_p$, it is computationally intractable to distinguish between (g, h, g^x, h^x) and (g, h, g^x, z), where $x \in_R Z_q$ and $z \in_R QR_p$.

Recall that the primary goal of the protocol is to resist off-line dictionary attacks by the two servers, where SS is a malicious adversary while CS is a semi-honest adversary. It is easy to see that outside attackers are no more powerful than SS in terms of uncovering U's password. We next examine the protocol against CS, SS and outside attackers, respectively.

(1) *Resistance to CS*

According to the adversary model, CS may eavesdrop on the communication channels to collect protocol transcripts. CS can obtain $B_1 = B/B_2 = g_1^{b_1} g_2^{\pi_1} \pmod p$ from $M4$. However, from B_1 alone, CS cannot obtain anything on π_1 in an information theoretic sense. What remains relevant to CS for off-line dictionary attacks are $[A = g_1^a$, $S_c = h((B/g_2^{\pi})^a)]$ and $[S_1 = A^{b_1}$, $B_1 = g_1^{b_1} g_2^{\pi_1}]$. The first pair is clearly no easier than $[A = g_1^a$, $S'_c = (B/g_2^{\pi})^a]$ for CS to handle. Note that $A = g_1^a \Rightarrow g_1 = A^{a^{-1}} \pmod p$ and $S'_c = (B/g_2^{\pi})^a \Rightarrow B/g_2^{\pi} = S_c'^{a^{-1}} \pmod p$. Under the DDH assumption, CS cannot distinguish between $[A$, $g_1 = A^{a^{-1}}$, S'_c, $B/g_2^{\pi} = S_c'^{a^{-1}}]$ and $[A$, $A^{a^{-1}}$, S'_c, $z]$, where $z \in_R QR_p$. This suggests that CS cannot get anything on π from the first pair. For the second pair, $B_1 = g_1^{b_1} g_2^{\pi_1} \Rightarrow B/g_2^{\pi_1} = g_1^{b_1} \pmod p$, and again under the DDH assumption, CS cannot distinguish between $[A$, $S_1 = A^{b_1}$, g_1, $B/g_2^{\pi_1} = g_1^{b_1}]$ and $[A$, A^{b_1}, g_1, $z]$. This shows that CS cannot learn anything on π_1 from the second pair. Consequently, as a semi-honest adversary, CS cannot launch effective off-line dictionary attacks.

It is important to note that in the above analysis, we have implicitly assumed that CS does not know $a = \log g_1^A$. However, were CS a malicious active adversary, such an assumption would no longer hold, since CS could simply impersonate U, choose a and compute $A = g_1^a \pmod p$. CS could also break the system if it were able to replace the original A from U with another one based on an a chosen by itself. In both cases, CS could find the password π by off-line dictionary attacks. To see this, consider the second pair where CS knows $a = \log g_1^A$ and the Diffie-Hellman quadruple $[A,\ S_1 = A^{b_1},\ g_1,\ B_2/g_2^{\pi_1} = g_1^{b_1}]$. It follows that $(B_2/g_2^{\pi_1})^a = (B_2/g_2^{\pi-\pi_2})^a = A^{b_1} = S_1$, so CS could try every possible password to determine the actual π. This explains at the technical level why CS is assumed to be a semi-honest adversary.

Observe further that CS relies on direct computation of $g_1^{a(b_1+b_2)} \pmod p$ to validate the authenticity of U, and the same thing is also exploited by SS and U to authenticate each other and negotiate a common session key. This suggests that if CS were a malicious adversary, it could establish a session key in the name of SS. This is another reason for CS being semi-honest.

(2) *Resistance to SS*

First, if behaving as a semi-honest adversary like CS, of help for SS in terms of off-line dictionary attack is $[A = g_1^a,\ S_c = h((B/g_2^{\pi})^a)]$ and $[S_2 = A^{b_2},\ B_2 = g_1^{b_2} g_2^{\pi_2}]$. Following a similar analysis as above, we can show that SS is unable to learn anything on either π or π_2 from the two pairs. What remains to consider is when SS launches active attacks, in which cases SS may behave arbitrarily such as impersonating U, and modifying and replacing messages. From the security analysis for CS, we know that if

SS replaces A coming from U with g_1^a based on its choice of a and if this is not detected by CS, SS can obtain π by off-line dictionary attack. Fortunately, different from the case of CS, this attack cannot succeed for the following reasons: S_2 is sent to SS in $M7$ only after CS has already decided on the validity of $S_c \overset{?}{=} h(S_1 A^{b_2})$; it is not possible for SS to change A while at the same time making $S_c \overset{?}{=} h(S_1 A^{b_2})$ pass CS's test. As a result, as an active attacker, SS is still not effective in off-line dictionary attacks.

(3) *Security to outside attackers*

While no more effective than SS in terms of dictionary attacks, an outside attacker could attempt to acquire the session key K established between U and SS as follows: (1) to impersonate any of U, SS and CS. Clearly this reduces to deriving any of π, π_1 and π_2 by off-line dictionary attack; (2) computing the value of $g_1^{a(b_1+b_2)} \pmod p$ from the protocol transcripts. Of help to this end are S_c, S_s, S_1 and S_2. Obviously inverting S_c and S_s is impossible if the underlying hash function is secure. On the other hand, since the communication channel between SS and CS is secret, the attacker cannot observe S_1 and S_2. It is clear that given only one of S_1 and S_2 does not help the attacker in computing $g_1^{a(b_1+b_2)} \pmod p$. This suggests that one-way secrecy of the channel between SS and CS in fact suffices to guarantee the security of our protocol.

Performance analysis: we next examine performance of our protocol. Let $|p|$ and $|h|$ denote the bit length of p and $h(.)$, respectively. The theoretical computation and communication costs of the protocol are given in Table 3.2. Since exponentiation computations predominate an entity's workload, we only count the number of exponentiations as the computation performance, and the digit following "/" denotes the number of exponentiations that can be computed off-line. Note that by leveraging the techniques in [60], $g_1^{b_1} g_2^{\pi_1} \pmod p$ and $g_1^{b_2} g_2^{\pi_2} \pmod p$ can be computed in a single exponentiation. In addition, $M1$ and $M2$ are not counted in the evaluation of communication cost.

	U	SS	CS
Computation (exponentiations)	3 / 2	2 / 1	2 / 1
Communication (bits)	$2\|p\| + 2\|h\|$	$6\|p\| + 3\|h\|$	$4\|p\| + \|h\|$

Table 3.2: Performance of the Preliminary Protocol.

Table 3.2 shows that our protocol is quite efficient in terms of both computation and communication, to all entities. Take U for example, U needs to calculate 3 exponentiations, where 2 are off-line; the communication cost for SS is also low, about $2|p| + 2|h|$ bits in total. As a result, our protocol can readily support wireless applications. It is also interesting to observe that the protocol technically supports a single central server with multiple service servers, since from Table 3.2, the workload (in both computation and communication) of the central server is low. Of course, with adequate funds, the security manager can always deploy a more powerful hardware for the central server.

Discussion: The preliminary protocol has two main weaknesses: first, it is clear that CS knows the session key K established between U and SS. While CS is semi-honest, this is not desirable; second, we have assumed a secret communication channel between SS and CS. We next present our final authentication and key exchange protocol by circumventing all these drawbacks.

3.2.3 The Final Authentication and Key Exchange Protocol

Our intuition to address the weaknesses in the preliminary protocol is that given the verification data on which CS depends to verify the authenticity of U, CS is unable to derive the secret U uses to verify the servers, which is also used by U and SS to compute the common session key.

Adversary model in the final protocol in the same as in the preliminary protocol, but no secret communication channel is assumed. Let P, p, q, g_1, g_2, g_3 and $h(.)$ be defined as in Table 3.1, and suppose U has already registered π_1 to SS and π_2 to CS as in the preliminary protocol, we outline the final protocol in Figure 3.6. Note that arithmetic operations associating with g_1, g_2 are performed modulo p, associating with g_3 are modulo P.

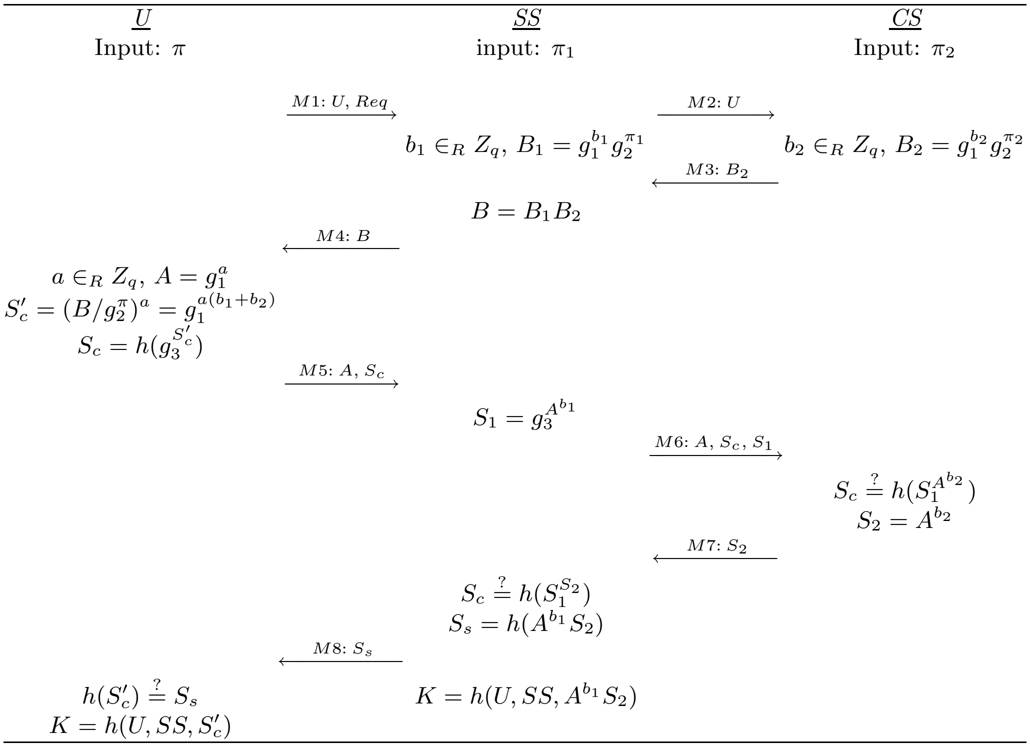

$\underline{U}$
Input: π
$\underline{SS}$
input: π_1
$\underline{CS}$
Input: π_2
$\xrightarrow{M1:\,U,\,Req}$
$\xrightarrow{M2:\,U}$
$b_1 \in_R Z_q,\ B_1 = g_1^{b_1} g_2^{\pi_1}$
$b_2 \in_R Z_q,\ B_2 = g_1^{b_2} g_2^{\pi_2}$
$\xleftarrow{M3:\,B_2}$
$B = B_1 B_2$
$\xleftarrow{M4:\,B}$
$a \in_R Z_q,\ A = g_1^a$
$S_c' = (B/g_2^{\pi})^a = g_1^{a(b_1+b_2)}$
$S_c = h(g_3^{S_c'})$
$\xrightarrow{M5:\,A,\,S_c}$
$S_1 = g_3^{A^{b_1}}$
$\xrightarrow{M6:\,A,\,S_c,\,S_1}$
$S_c \overset{?}{=} h(S_1^{A^{b_2}})$
$S_2 = A^{b_2}$
$\xleftarrow{M7:\,S_2}$
$S_c \overset{?}{=} h(S_1^{S_2})$
$S_s = h(A^{b_1} S_2)$
$\xleftarrow{M8:\,S_s}$
$h(S_c') \overset{?}{=} S_s$
$K = h(U, SS, S_c')$
$K = h(U, SS, A^{b_1} S_2)$

Figure 3.6: The Final Authentication and Key Exchange Protocol Using Password.

Specifically, U sends his identity as well as a service request Req to SS in $M1$, in order to initiate a service request. SS first relays the user ID to CS in $M2$, and then selects a random number $b_1 \in_R Z_q$, and use his password share π_1 to compute

$B_1 = g_1^{b_1} g_2^{\pi_1} \pmod{p}$. Upon receiving $M2$, CS chooses a random number $b_2 \in_R Z_q$, which is used together with π_2 to calculate $B_2 = g_1^{b_2} g_2^{\pi_2} \pmod{p}$. CS then sends B_2 in $M3$ to SS. Upon reception of B_2, SS computes and sends $B = B_1 B_2 \pmod{p}$ to U in $M4$. With B received, U selects randomly $a \in_R Z_q$, and in turn computes $A = g_1^a \pmod{p}$, $S_c' = (B/g_2^{\pi})^a = g_1^{a(b_1+b_2)} \pmod{p}$ and $S_c = h(g_3^{S_c'} = g_3^{g_1^{a(b_1+b_2)}} \pmod{P}))$. Afterwards, U sends A, S_c to SS in $M5$. Upon getting $M5$, SS first computes $S_1 = g_3^{A^{b_1}} \pmod{P}$, and then sends $(A,\ S_c,\ S_1)$ to CS in $M6$. Upon reception of $M6$, CS tests $S_c \overset{?}{=} h(S_1^{A^{b_2}} \bmod P)$: if it holds, CS is assured of the authenticity of U, and continues the protocol by computing and sending $S_2 = A^{b_2} \pmod{p}$ to SS in $M7$; otherwise, CS aborts the protocol.

Next, upon reception of S_2, SS checks $S_c \overset{?}{=} h(S_1^{S_2} \pmod{P})$: if it holds, SS is convinced of the authenticity of U, and computes and sends $S_s = h(A^{b_1} S_2 \pmod{p})$ to U in M_8; otherwise, SS aborts the protocol. So far, both servers have verified U. Upon receiving M_8, U checks $h(S_c') \overset{?}{=} S_s$: if it holds, U has validated the server, and computes a session key $K = h(U, SS, S_c')$; otherwise, U aborts the protocol. SS also computes a session key $K = h(U, SS, A^{b_1} S_2)$.

Correctness: For the purpose of verifying U, CS needs to check $S_c \overset{?}{=} h(S_1^{A^{b_2}} \pmod{P})$, and SS needs check $S_c \overset{?}{=} h(S_1^{S_2} \pmod{P})$. To make the checks work, it must hold that $g_3^{(g_1^{a(b_1+b_2)} \bmod p)} \pmod{P} = g_3^{(g_1^{ab_1} \bmod p)(g_1^{ab_2} \bmod p)} \pmod{P}$. However, normally $g_1^{a(b_1+b_2)} \bmod p \neq (g_1^{ab_1} \bmod p)(g_1^{ab_2} \bmod p)$, but it hods that $g_1^{a(b_1+b_2)} \bmod p = (g_1^{ab_1} \bmod p)(g_1^{ab_2} \bmod p) \pmod{p}$. As $g_3 \in QR_P$, and g_3 is of order p, the above checks thus hold.

Security: Based upon the security analysis of the preliminary protocol, we only need to focus on where the two protocols differ. It is clear that the introduction of arithmetic operations associating with g_3 makes it no easier for SS and CS in terms of off-line dictionary attacks. We remain to consider security against outside attackers. Similarly, outside attackers are no more powerful than SS with respect to dictionary attacks. We next see whether they can derive the common session key established between U and SS, as now the communication channel between CS and SS is open. As a result, compared to the preliminary protocol, an outside attacker can additionally get $S_1 = g_3^{A^{b_1}} \pmod P$ and $S_2 = A^{b_2} \pmod p$. The attacker has to obtain $A^{b_1} \pmod p$ in order to recover the common session key K. However, the attacker is not able to get A^{b_1} from $S_1 = g_3^{A^{b_1}} \pmod P$, which is equivalent to computing the discrete log of S_1. Notice that for a similar reason, CS cannot derive K either. We thus have addressed the weaknesses in the preliminary protocol.

Efficiency: let $|p|$, $|h|$ and $|P|$ denote the bit length of p, $h()$ and P, respectively, the theoretical computation cost and communication cost are listed in Table 3.3.

	U	SS	CS
Computation (exponentiations)	4/2	4/1	3/1
Communication (bits)	$2\|p\| + 2\|h\|$	$5\|p\| + 3\|h\| + \|P\|$	$3\|p\| + \|h\| + \|P\|$

Table 3.3: Performance of the Final Protocol.

From the table, the final protocol obtains similar performance as the preliminary protocol (shown in Table 3.2).

To test the actual efficiency, we implemented the protocol using Visual C++: the communication module was coded upon Winsock 2 (http://www.sockets.com/winsock2.htm), and the cryptographic operations were coded upon OpenSSL (http://www.openssl.org/);

we ran the experiments on PCs with Pentium 4 processors (2.1 GHz), 521M RAM, and Windows XP OS; in our implementation, all computations are done on-line, without performing any off-line arithmetic operations in advance (although as shown in Table 3.2, some exponentiation operations can be computed off-line.). Our experiments use $|p| = 1024$, and the results show that the average time for executing the protocol is around 0.3 seconds.

3.2.4 Features of the Two-server Password System

We summarize features of our proposed two-server password system.

- At the architectural level, the two-server system exploits the organizational trust structure and more precisely the security manager, thereby coherent and compatible with other user authentication techniques and modes.
- At the technical level, the single point of vulnerability inherent in traditional single-server systems is eliminated. As a result, neither the central server nor the service servers can compromise user passwords by means of off-line dictionary attack.
- As the central sever is hidden from the public, the chance for it under attacks is substantially minimized, thereby increasing the overall security of the whole system.
- A user can use the same password to register to different services (service servers) by varying the two shares of the password. This avoids a big inconvenience in traditional password systems, where a user has to memorize different passwords for different applications.

- Compromising solely the service servers does not lead to the compromise of user passwords. Therefore, the units or departments managing service servers are relieved to some extent from strict security management, and they can dedicate their limited expertise and resources to enhancing service provision to users.
- Users are afforded to assume the higher credit of the organization, while engaging business with individual units or departments of the care organization.
- Since user authentication must involve the central server, the organization is actually provided a way to monitor the affiliating units or departments, if desired.

3.2.5 Related Work on Password Authentication

While password is one of the earliest user identification and authentication techniques, there is resurgent effort in developing password systems that are resilient against off-line dictionary attacks recently. It is a proven fact that *public key techniques* (e. g., exponentiations in a multiplicative group) are absolutely necessary to make such systems secure against off-line dictionary attacks, whereas the involvement of public key cryptosystems (e. g., public key encryption and digital signature schemes) is not essential [88]. This observation differentiates two separate approaches to the development of secure password systems: combined use of password and public key cryptosystem, and password-only approach. The former takes into account the asymmetry of capabilities between users and servers, so a user only uses a password while the server has a public/private key pair at its disposal. Examples of such public key-assisted password authentication systems include [34, 76, 88]. With no exception, the use of public keys entails the deployment and maintenance of a PKI for public key certification, and adds to users the burden of checking key validity. To eliminate this drawback, password-only authenticated key exchange (PAKE) protocols have been extensively studied (e.g., [20,31,32,37,112,113]). The PAKE protocols do not involve any public key cryptosystem and therefore are much more attractive for real world applications. We believe that any use of public key(s) in a password system should be avoided, since otherwise the benefits brought by the use of password would be counteracted to a great extent.

Most of the existing password systems (including PAKEs) were designed over a single-server model, where each user shares a password or some PVD with a single authentication server. While PAKE protocols are sufficiently robust against off-line attacks mounted by outsiders, they are by no means resilient to off-line dictionary attacks initiated at the server side, e.g., in the event of server break-ins by outsiders or misbehavior by unscrupulous system administrators. To address this problem, password systems based on multi-servers were proposed. The principle of the multi-server model is distributing the password/PVD database as well as the verification functionality that are originally imposed upon a single server to multiple servers in order to eliminate the single point of vulnerability. As such, without compromising multiple servers, an attacker is bound not to be effective in off-line dictionary attacks. The system in [69], believed to be the first multi-server password system, splits a short password among multiple servers. However, the servers in [69] need to use public keys. An improved version of [69] was proposed in [101] which eliminates the use of public keys by the servers. Further and more rigorous extensions were due to [134] and [150], where the former built a t-out-of-n threshold PAKE protocol and provided a formal security proof under the random oracle model [38], and the latter presented two provably secure threshold

PAKE protocols under the standard model. While the protocols in [134] and [150] are theoretically significant, they have low efficiency and high operational overhead. In these multi-server password systems, the servers are equally exposed to the users and a user has to communicate in parallel with several or all servers for authentication. This may either cause problems to resource constrained mobile devices such as hand phones and PDAs, or has compatibility problem as most existing systems are single-server systems. Moreover, multi-server systems are also subject to the so called common-mode failures in practice, that is, if an attacker knows how to break one server, highly likely he can break others [158].

The password system most closely related to ours is the two-server system recently proposed by Brainard *et al.* [27], where one server exposes itself to users and the other is hidden from the public. While this two-server setting is efficient, it is not a password-only system: both servers need to have public keys to protect the communication channels from users to servers. As we have stressed earlier, this makes it difficult to fully enjoy the benefits of a password system. In addition, the system in [27] only performs unilateral authentication and relies on the Secure Socket Layer (SSL) to establish a session key between a user and the front-end server. Subsequently, Yang *et al.* [187] extended and tailored this two-server system to the context of federated enterprises, where the back-end server is managed by an enterprise headquarter and each affiliating organization operates a front-end server. Nevertheless, the system in [187] still does not completely avoid the use of public keys, as the back-end server holds a public key. Our proposed system continues this line of research on the two-server architecture, whereas we adopt a very different method at the technical level, which gives rise to a password system requiring no public key whatsoever. Furthermore, our system has no compatibility problem, since from users' point of view, users need to communicate with only one server.

3.3 Concluding Remarks

We suggested building a unified trust infrastructure for health care organizations that accommodates various user authentication techniques and modes. The organizational trust infrastructure enables a health care organization to achieve other aspects of data security within or beyond organizational boundaries. In particular, each organization establishes an organizational security manager that manages security related issues; user authentication techniques are then built around the security manager. To tailor password authentication to the organizational trust infrastructure, we proposed a novel two-server password authentication system that enlists the security manager for operating a back-end server. We must point out that the proposed two-server password authentication system is a generic user authentication technique, and it clearly has applications in other contexts than health care systems. What relevant here is that it makes password authentication unified with other user authentication techniques under the proposed trust infrastructure for health care systems.

CHAPTER 4

Anonymous Remote Login Scheme for Health Care Services

In Chapter 3, we have discussed user authentication in health care, and in particular, we suggest building a unified trust infrastructure for health care organizations that accommodates various user authentication techniques and modes. With the organizational trust infrastructure in place, we are ready to address security issues beyond organizational boundaries. In this chapter, we present a remote login scheme that allows a health care organization, e.g., a hospital, to provide a certain service to users, such that users can access the service from off-site locations in an anonymous manner. More specifically, different accesses by the same user cannot be linked by outside attackers. This provides a useful alternative to the user authentication techniques including the two-server password authentication system in Chapter 3, when users are willing to achieve strong user privacy in the login process, where user identification occurs.

The development of public networks, especially Internet, enables health care organizations to allow employees, physicians and patients to access clinical services and information from off-site locations. The services provided by health care organizations can take various forms, e.g., a WWW dental consultation service to patients, or a FTP service provided by the radiological department of a hospital to physicians belonging to that department. In such systems, to access a service from a remote-site location, a user (a physician or a patient) must initiate a login process with the organization that provides the service, whereby they first identify and verify each other, and upon a successful mutual authentication they then negotiate a shared session key for the protection of successive data communication between them. Essentially, this in general falls into the class of user authentication discussed in Chapter 3, whereas as it will be clear shortly, we are concerned with more features than the basic user authentication functionality. Numerous work has been dedicated to user authentication and key exchange in the login process. For example, two widely used systems, SSL (Secure Sockets Layer) [70] and Kerberos [107] are among the effort, and SSL uses public key cryptosystems while Kerberos relies on symmetric key cryptosystems; password authentication systems (as we reviewed in Chapter 3) belong to an alternate class in the sense that users use passwords for user authentication.

In health care systems, users have a privacy concern in the remote login process, due to that much sensitive information pertaining to users such as individual professions and health situations is conveyed from the services (even without knowing the content) they are accessing. For example, if a user frequently visits the website of a dental service, most probably the user has dental problem. Furthermore, when widely collected and compiled, information of this kind might be abused for, e.g., marketing purposes [22]. As such, disclosing the login information clearly harm individual privacy. Ideally, users should be kept anonymous in the login process while the functionality of user identification and key exchange is still enabled. More precisely, only the valid parties at both ends of the communication could identify each other, and outside attackers should not be allowed to learn *who* are in the login process. We further desire a stronger notion of user privacy, *unlinkability.* That is, different login sessions by the same user cannot be linked. Unfortunately, most of the existing approaches, including the aforementioned techniques, do not take these issues into consideration. Let us first take SSL for instance, it adopts X.509 certificate [186] for the purpose of user identification, so the identifying information of an individual is explicit in the login process. The same applies to the password authentication systems where users must input IDs together with passwords in order for user authentication. Kerberos has similar weakness. As a result, by simply eavesdropping on the login traffic, an outside attacker can readily discern who is accessing the service.

To address this problem, we present a remote login scheme, by which users can access health care services from off-site locations in an anonymous (and unlinkable) manner. We develop our scheme by rectifying a serious weakness existing in a scheme proposed by Wu and Hsu [182] that similarly attempted to ensure user privacy in the login process.

4.1 An Anonymous Remote Login Scheme

Since the login scheme deals with a health care organization providing a service to its affiliated users such as physicians and patients, it is built upon the organizational trust infrastructure, with the security manager of the organization managing security related matters. We start by giving a high level description of the overall system.

4.1.1 High Level Description

Three types of participants are involved in the system, and they are defined as follows.

- **Service providers**: A service provider is the party that provides a certain service to its affiliated users. A service provider may be a department or a unit within a care organization or the organization itself. An organization may support multiple service providers, each providing a certain service. We denote a service provider as SP for short,

- **Users**: Users are the parties willing to anonymously access the service provided by the service provider they are affiliated to. For example, users may be dentists or patients of the dental department in a hospital. A user is denoted as U.
- **Security Manager**: Security manager is the *security manager* of the care organization as established in Chapter 3, trusted by users and service providers. The security manager is responsible for setting up and publishing system parameters and issuing a secret token to each user and each service provider. Users and service providers use their respective secret tokens in the login process, identifying each other and negotiating a common session key. The security manager works as an off-line party, only getting evolved in a *registration* procedure.

The system works in the following procedures, as outlined in Figure 4.1. The security manager first sets up system parameters in a *system setup* procedure. In an off-line *registration* procedure, each service provider and each user register to the security manager, who issues a secret token to each registrant. To request service from the service provider, a user initiates a *login* session, where the user and the service provider authenticate each other and negotiate a common session key for establishing a secret communication between them.

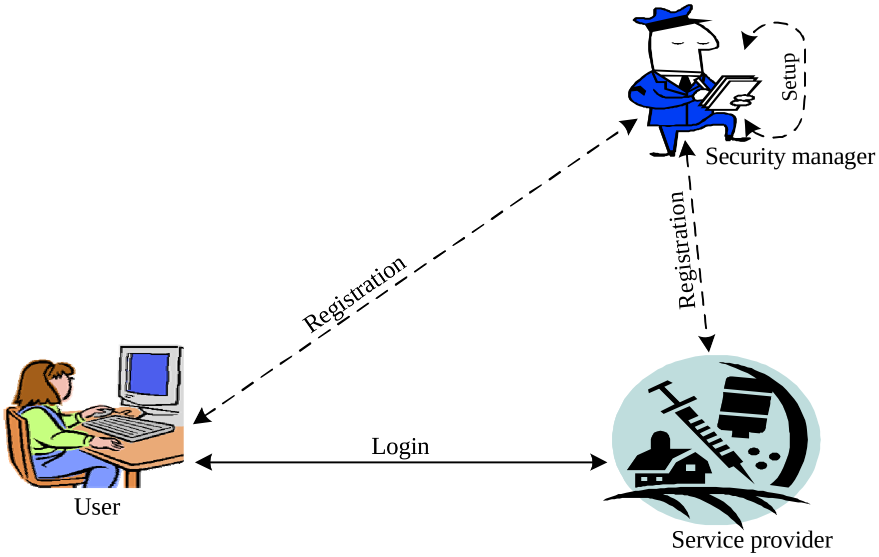


Figure 4.1: Procedures of An Anonymous Login System.

4.1.2 Security Requirements

Let $\mathcal{U}$ be the whole set of users that had registered to the security manager, and $\mathcal{R}$ be the whole set of login transcripts. Besides common requirements on user authentication systems, e.g., resistance to impersonation, etc., we further impose the following security requirements upon the anonymous login system.

1. **User Anonymity** Given $R \in \mathcal{R}$, it is computationally infeasible for a probabilistic polynomial-time outside adversary to decide $U \in \mathcal{U}$ that R belongs to.
2. **User Unlinkability** Given $R_1, R_2 \in \mathcal{R}$, it is computationally infeasible for a probabilistic polynomial-time outside adversary to decide whether R_1 and R_2 belong to the same user in $\mathcal{U}$. This actually suggests that an outside adversary is unable to link users based on the content of login transcripts. However, as the login scheme is most probably for Internet services, an attacker can instead directly check the IP addresses of the login transcripts for the purpose of linking

users. We therefore must assume that the anonymous login scheme is built over a type of networks such as Anonymizer [75] and Mix [44] that circumvent such a kind of IP address linking .

3. **Secrecy of Tokens** Neither a user nor the service provider can learn the secret token of each other.

Definition 4.1: A login system is *anonymous* if it satisfies all of the above requirements.

4.1.3 The Scheme

We are now ready to elaborate on our construction of an anonymous login system.

1. *System Setup*

To set up system parameters, the security manager does the following:

- Chooses two large primes p and q, and computes $n = pq$; selects e and computes d such that $ed = 1 \bmod \phi(n)$, where $\phi(n) = (p-1)(q-1)$. Note that e should be sufficiently large, e.g., 160 bits.
- Chooses an element $g \in Z_n^*$, which is the generator of both Z_p^* and Z_q^*, that is, $g \in QR_n$.
- Picks a symmetric-key cryptosystem such as AES [4], whose encryption function and decryption function under the secret key K are $E_K(.)$ and $D_K(.)$, respectively. A cryptographic hash function $h(.)$ is also selected.
- Publishes $(e,\ n,\ g)$ as public system parameters and keeps $(d,\ p,\ q)$ secret.

2. *Registration*

Suppose a user U or a service provider SP has already identified to the security manager, e.g., a user shows her/his identity card or driver license. Through a secure channel, the security manager issues a secret token S_i to the registrant. S_i is computed as:

$$S_i = (ID_i)^d \pmod{n} \tag{4.1}$$

where ID_i is the identity of U or SP, and we suppose $ID_i \in QR_n$ are well-formed, e.g., email address or social security number [1]. Note that for users, the secret tokens may be carried directly by users' smart cards, as we have assumed that each patient or medical personnel has a personal smart card at her/his disposal; for service providers, as they reside in the organization, it is not difficult for them to identify to and in turn establish a secure channel with the security manager.

3. *Login*

To request service from a service provider SP, user U initiates a login session by executing the protocol in Figure 4.2 (arithmetic operations are performed modulo n), where U and SP identify each other and negotiate a shared session key. Note that U holds a secret token $S_u = ID_u^d \pmod{n}$ and SP holds $S_{sp} = ID_{sp}^d \pmod{n}$.

The protocol works as follows. User U initiates a login session by sending a service request *Req* to the service provider SP in $M1$. Upon reception of the request, SP

[1] Recently, [189] gave attacks on this registration step, indicating that an attacker can forge S_i. However, their attacks are only possible if the security manager allows arbitrary IDs for registration or allows a user to register using multiple IDs. This is clearly not true in practice: registration in any system is a highly observed and stringent procedure, and users are strictly required to present and prove their real identities. Therefore, the attacks in [189] are not practical, and will never happen in real systems.

$$\begin{array}{ccc}
\underline{U} & & \underline{SP} \\
\text{input: } S_u & & \text{input: } S_{sp} \\
 & \xrightarrow{M1\text{: } Req} & \\
 & & k \in_R Z_n, z = g^k S_{sp}^{-1} \\
 & \xleftarrow{M2\text{: } z} & \\
a = z^e ID_{sp} & & \\
t \in_R Z_n, K = a^t & & \\
x = g^{et} & & \\
s = g^t S_u^{h(x,\, T)} & & \\
y = E_K(ID_u) & & \\
 & \xrightarrow{M3\text{: } x,\, s,\, y,\, T} & \\
 & & K' = x^k \\
 & & ID_u = D_{K'}(y) \\
 & & xID_u^{h(x,\, T)} \overset{?}{=} s^e
\end{array}$$

Figure 4.2: An Anonymous Login Protocol.

chooses $k \in_R Z_n$ and computes

$$z = g^k S_{sp}^{-1} \pmod{n} \tag{4.2}$$

using his secret token S_{sp}. z is then sent to U in $M2$. Upon receiving z, U chooses $t \in_R Z_n$ and does the following computations.

$$\begin{aligned}
a &= z^e ID_{sp} \pmod{n} && (4.3)\\
K &= a^t \pmod{n} && (4.4)\\
x &= g^{et} \pmod{n} && (4.5)\\
s &= g^t S_u^{h(x,\, T)} \pmod{n} && (4.6)\\
y &= E_K(ID_u) && (4.7)
\end{aligned}$$

where T is the current timestamp. Note that K will be the common session key. Afterwards, U sends x, s, y, T to SP in $M3$. When SP receives the message, he first checks the timestamp T: if it is not within a pre-defined window, SP aborts the protocol; otherwise, the protocol continues. For the purpose of user identification, SP first computes $K' = x^k \pmod{n}$, then proceeds to decrypt y as $ID_u = D_{K'}(y)$. SP then checks whether ID_u is a legitimate user: if ID_u is not a legitimate user, SP simply aborts the protocol; otherwise, SP continues to verify

$$xID_u^{h(x,\,T)} \overset{?}{=} s^e \pmod{n} \tag{4.8}$$

If the equation holds, then SP has validated U and grants the requested service, and K' will be the common session key; otherwise, request is rejected. We point out the validity of the common session key K and K' is automatically verified, and no additional round of interaction between U and SP is needed. To see this, if U and SP compute different keys ($K \neq K'$), the possibility of making Equation 4.8 hold is negligible. This is an added feature of our system.

We next check the correctness of Equation 4.8. Raising s to the power e, it yields

$$\begin{aligned} s^e &= g^{et} S_u^{eh(x,\,T)} \pmod{n} && \text{by Equation 4.6} \\ &= xID_u^{deh(x,\,T)} \pmod{n} && \text{by Equation 4.1 and 4.5} \\ &= xID_u^{h(x,\,T)} \pmod{n} && \text{by RSA} \end{aligned} \tag{4.9}$$

4.1.4 Security Discussions

In this subsection, we examine security of our proposed scheme, and analyze how the above construction satisfies the earlier security requirements.

First of all, since a timestamp is involved in our protocol, replay attack is thus prevented. For use of timestamp, we assume synchronization of clocks among the participants is not a problem. Analyses that follow are based mainly on the following assumptions.

> *RSA assumption* [153]: Let $N = pq$ and $\gcd(e, \phi(N)) = 1$, where p and q are unknown large primes. Given $y \in Z_N$, it is computationally intractable to derive x such that $y = x^e \pmod N$ with the knowledge of e and N. RSA assumption is eventually reduced to the hardness of the factorization of N.
>
> *DH assumption* [56]: Let $p = 2q + 1$ be a large prime, where q is also a large prime, and g be a generator of Z_p^*. Given $g^x \pmod p$ and $g^y \pmod p$, it is computationally intractable to compute $g^{xy} \pmod p$. The DH assumption results from the hardness of discrete logarithm: given y, it is computationally intractable to compute x such that $y = g^x \pmod p$. This assumption is believed to still hold for a composite modulus (see for example [63, 128]).

For clarity of security discussion, we classify possible attacks to the scheme into two classes: attacks to user U, and attacks to service provider SP. We next in turn examine each of them.

(1) *Attacks to U*

Attacks to U could be due to either SP or outside attackers. We first check attacks by SP. Attacks to U by SP are restricted to derive the secret token S_u of U, and in turn impersonate U to request services from other service providers. First, from $s = g^t S_u^{h(x,T)} \pmod n$ clearly SP is unable to get S_u, since it at least involves computing discrete logarithm even g^t is known. Instead, SP may proceed in the following steps.

- Attempts to obtain g^t from $x = g^{et} \pmod{n}$ in Equation 4.5, in conjunction $K' = g^{ekt} \pmod{n}$.
- Computes $S_u^{h(x,\, T)} = x/g^t \pmod{n}$.
- Given two prior sessions at time T_1 and T_2 such that $\gcd(h(x_1,\, T_1),\, h(x_2,\, T_2)) = 1$[2], SP finds m_1 and m_2 such that $m_1 h(x_1,\, T_1) + m_2 h(x_2,\, T_2) = 1$.
- Finally, computes $S_u = S_u^{m_1 h(x_1,\, T_1)} S_u^{m_2 h(x_2,\, T_2)} \pmod{n}$.

However, computing $g^t \pmod{n}$ from x or/and K' is computationally intractable, according to the RSA assumption.

To impersonate U without deriving S_u, SP is faced to forge x in Equation 4.5 and s in Equation 4.6 and make s pass the test in Equation (4.8), i.e., $s^e = xID_u^{h(x,\, T)} \pmod{n}$. This reduces to the following cases: (a) SP first determines a random s, and then tries to compute x. This is intractable from the DH assumption and provided that the one-way hash function is secure; (b) SP first determines x and then attempts to compute s, which is computationally intractable from the RSA assumption; (c) SP chooses a random number r and sets $x = ID_u^r \pmod{n}$, so $xID_u^{h(x,\, T)} = ID_u^{(r+h(ID_u^r, T))} \pmod{n}$. If SP can find a w and set $s = ID_u^w \pmod{n}$, such that $ew = r + h(ID_u^r, T)$ holds, then SP is successful. This is equivalent to determining r such that $r = h(ID_u^r, T) \pmod{e}$. According to the well-known birthday paradox [157], such a r can be found within $\sqrt{e}$ trials. Therefore, we stipulate that e must be large enough.

[2] The probability of $\gcd(h_1,\, h_2) = 1$ is expected quite large, where h_1 and h_2 are two random hash values. We give a rather rough estimation as follows: suppose $|h(.)| = l$, then the number of primes less than 2^l is about $n = 2^l/ln2^l$. Further, we suppose each hash value has κ prime factors in average. Consequently, The probability Pr of $\gcd(h_1,\, h_2) = 1$ can be computed as $Pr = \binom{n}{\kappa}\binom{n-\kappa}{\kappa}/\binom{n}{\kappa}\binom{n}{\kappa} = \binom{n-\kappa}{\kappa}/\binom{n}{\kappa}$. Considering $\kappa \ll n$, Pr will be quite large.

We next consider attacks by outside attackers. Besides launching similar attacks as PS, an outside attacks may be additionally to break the user anonymity by figuring out who is in conversation with the service provider SP by observing the data traffic along the communication or to link users. Clearly, outside attackers are no more powerful than SP with respect to impersonation attacks or deriving the secret token. For attacks to break user anonymity and user unlinkability, as all of x, s, y, T in $M3$ vary with sessions, an attacker is left with no clue to link data. The only way to break user anonymity (and further user unlinkability) is to compute K (or K', which is the same as K). This is no easier than to break the DH assumption, provided that the underlying symmetric key cryptosystem is secure.

(2) *Attacks to SP*

Attacks to SP could be due to either U or outside attackers, whereas they are restricted to the same objectives, i.e., to derive S_{sp} or impersonate SP. Clearly, outside attackers gain no more advantages than U in such attacks. For simplicity, we only consider attacks by U.

To derive S_{sp}, U can compute $a = g^{ek} \pmod{n}$ by z in Equation 4.3. However, to acquire S_{sp} from z in Equation 4.2, U needs $g^k \pmod{n}$. From the RSA assumption, we know it is impossible to compute $g^k \pmod{n}$ from e and $(g^k)^e \pmod{n}$. Knowing $K' = g^{ekt} \pmod{n}$ does not help either in this aspect as we discussed earlier. We thus conclude that deriving S_{sp} is not possible by U.

We proceed to examine how U impersonates SP. If U, in the name of SP, can successfully share a common session key with another user U', then the impersonation attack is deemed successful. There exist two ways for U to impersonate SP. (1) On intercepting the service request from U', without knowing S_{sp}, U simply chooses another random number m and computes $z = g^k m \pmod{n}$ in Equation 4.2. In such a case, K computed by U' in Equation 4.4 is $K = a^t = g^{ekt} m^{et}/ID_{sp}^t = g^{ekt}(m^e/ID_{sp})^t \pmod{n}$. U is then faced to compute K' (the same as K) with the knowledge of $x = g^{et} \pmod{n}$. Although U can compute $g^{ekt} \pmod{n}$ from x, he cannot compute $(m^e/ID_{sp})^t \pmod{n}$ without knowing t. (2) Another way for U to impersonate SP is to exploit a z of a past session from SP (we suppose in that session, SP chose k_p and U chosen t_u, so $z = g^{k_p}.S_{sp}^{-1} \pmod{n}$ and $K = g^{ek_p t_u} \pmod{n}$). To do so, U chooses a random number k' and compute $z' = g^{k'}.z \pmod{n}$ and sends z' to U' upon intercepting the service request from U'. In such a case, $\tilde{K}$ computed by U' in Equation 4.4 is $\tilde{K} = a^t = g^{et(k_p+k')} = g^{etk'}.g^{etk_p} \pmod{n}$. To compute $\tilde{K}'$ the same as $\tilde{K}$, although U can compute $g^{etk'} \pmod{n}$ from $x = g^{et} \pmod{n}$ in Equation 4.5 sent by U', he cannot compute $g^{etk_p} \pmod{n}$ from x and $g^{et_u k_p} \pmod{n}$, according to the RSA assumption as well as the DH assumption.

We conclude security discussions with the following claim:

Claim 4.1: *Our construction is an anonymous login system, satisfying the earlier security requirements.*

4.1.5 Performance Analysis and Implementation Results

For the purpose of performance analysis, we present a comparison between our proposed scheme with the Wu-Hsu scheme [182] (we shall give a brief review of the Wu-Hsu scheme shortly) in terms of both theoretical computational and communication performance. As usual, we only count the number of exponentiations as the computation performance, and the digit following "/" denotes the number of exponentiations that can be computed beforehand. In addition, as noted in Chapter 3, the computation of $g^t S_i^{h(x,T)} \pmod{n}$ needs only one exponentiation by the techniques in [60]. Let $|n|$, $|E|$ and $|T|$ denote the bit length of n, $E(.)$ and the time stamp T, respectively, the comparison results are listed in Table 4.1.

	Communication Cost	Computational Cost for U	Computational Cost for SP
Our scheme	$3\|n\| + \|E\| + \|T\|$	4/1	4/1
The Wu-Hsu Scheme	$3\|n\| + \|T\|$	4/1	4/1

Table 4.1: Performance Comparison between Our Scheme and the Hu-Hsu Scheme.

From Table 4.1, we can see that our scheme has similar performance as the Wu-Hsu scheme in terms of both communication and computation.

We also implemented a prototype of the protocol. The source code was written in Visual C++, and in particular, the communication module was coded using Winsock 2 (http://www.sockets.com/winsock2.htm), and the cryptographic operations were coded upon OpenSSL (http://www.openssl.org/). The experiments were run on two PCs with Pentium 4 processors (2.1 GHz), 521M RAM, and Windows XP OS. Our implementation treated all computations on-line, without doing any off-line arithmetic operations (so we can expect an enhancement in the performance of a practical deployment of the protocol through moderate implementation optimizations). We use $|n| = 1024$, and the average running time for the protocol is about 0.2 seconds.

4.1.6 Features of the Login Scheme

The scheme we propose has the following features.

- Clearly, it achieves user privacy against the public. This is the primary objective we are interested in. We point out that some user authentication techniques introduced in Chapter 3, e.g., anonymous credential and group signature, achieve similar properties, but they are much more computationally expensive.
- Each user only needs to maintain one secret (token), for accessing different service providers.
- Users do not need to hold individual certificates associating with the secrets they use. This differs radically from other user authentication techniques that achieve user privacy such as anonymous credential and group signature discussed in Chapter 3. In this sense, our proposal bears some similarities with the identity based cryptosystems (e.g., [154]).
- No user secret (e.g., password) is required to be hosted by service providers.

We further see other features of our scheme. In essence, users and service providers rely on long secrets (secret tokens) for authentication. A user is not expected to memorize such long secrets. This differs radically from the use of passwords as in Chapter 3, so mobility of uses seems affected. In fact, recall that we have assumed each user in health care systems has a personal smart card, so mobility of users is supported as users can carry their secret tokens by the smart cards.

Denial-of-Service (DoS) is a kind of attacks aiming at blocking regular service accessing, rather than acquiring secret content (e.g., secret keys or plaintexts). In nature, DoS attacks are hard to resist yet common for on-line applications. In our proposed protocol (referring to Figure 4.2), the response z in $M2$ to a user request can be computed beforehand by the service provider SP, so at this stage the protocol does not suffer from

DoS. Upon reception of $M3$, SP first needs to do an exponentiation (compute K') plus a decryption (compute ID_u). Note that to make K' in a correct form, which in turn makes the decryption result in a correct form, an DoS attacker has to do two exponentiations (compute a and K) and an encryption (compute y). Conversely, a DoS attacker sending random messages in $M3$ only costs SP one exponentiation computation. Note also that, this computation of two exponentiations plus an off-line exponentiation (compute x) is the minimum cost that a DoS attacker has to spend in order to lead SP to the more expensive test (to verify Equation 4.8). Considering these facts, our protocol is thus well resilient to DoS attacks, a feature important yet hard to achieve in anonymous systems

4.2 Related Work and An Attack to the Wu-Hsu Scheme

Lee and Chang [115] proposed an interesting scheme for user identification and key distribution with user anonymity. However, Wu and Hsu [182] pointed out that the Lee-Chang scheme suffers from some vulnerabilities. The main one is that since the Lee-Change scheme only implemented one way authentication of users to the service provider, an attacker can easily impersonate the service provider. Wu and Hsu [182] further proposed a more efficient scheme for the same purposes by assuming a similar system setting. Unfortunately, as we shall demonstrate shortly, while the limitations of the Lee-Chang scheme were rectified in the Wu-Hsu scheme, a new serious weakness arises. In particular, the service provider can obtain a user's secret token at the end of the login when user identification and key negotiation are accomplished. This will definitely impair the interest of users, enabling the service provider to freely impersonate a user in requesting services from other providers. We must stress that in contrast to password, a user's secret token is a long-term strong credential, intended for identification in different applications. In what follows, we briefly review the Wu-Hsu scheme, followed by an attack.

The Wu-Hsu scheme assumes similar system participants, procedures and parameters as our scheme (the two schemes differ the in login procedure). Consequently, by *registration* each user U has a secret token $S_u = ID_u^d \pmod{n}$, and each service provider SP holds a secret token $S_{sp} = ID_{sp}^d \pmod{n}$. The login procedure of the Hu-Hsu scheme is outlined in Figure 4.3.

$$
\begin{array}{ccc}
\hline
\underline{U} & & \underline{SP} \\
 & \xrightarrow{M1:Req} & \\
 & & k \in Z_n, z = g^k S_{sp} \\
 & \xleftarrow{M2:z} & \\
\begin{array}{c} a = z^e/ID_{sp} \\ t \in Z_n, x = S_u h(a^t||T) \\ y = g^{et} \\ K = a^{tx} \end{array} & & \\
 & \xrightarrow{M3:x,\, y,\, T} & \\
 & & \begin{array}{l} \text{check } T \text{ and verify} \\ ID_u \overset{?}{=} (x/h(y^k||T))^e \\ \qquad K = y^{kx} \end{array} \\
\hline
\end{array}
$$

Figure 4.3: The Anonymous Login Protocol of the Hu-Hsu Scheme.

Based on our scheme, it is not difficult to understand the Hu-Hsu scheme, and we refer interested readers to [182] for a detailed introduction as well as security analysis. We next demonstrate a serious weakness in the Wu-Hsu scheme, which allows the service provider to freely impersonate the users who had ever requested services. This happens because the service provider can obtain the secret token of a user after a successful execution of the anonymous login protocol as shown in Figure 4.3. To see this, from x,

y and T, the data sent to the service provider SP by the user U, SP can first compute $h' = h(y^k||T) = h(g^{ket}||T) \pmod n$ with the k he has chosen, and then compute $x/h' = S_u h(a^t||T)/h' = S_u h(g^{ket}||T)/h(g^{ket}||T) = S_u \pmod n$. This is actually why the test $ID_u \overset{?}{=} (x/h(y^k||T))^e \pmod n$ by SP holds, which is eventually reduced to $ID_u = S_u^e$ $\pmod n$.

4.3 Concluding Remarks

We proposed a remote login scheme for health care services, achieving strong user privacy (user anonymity and user unlinkability). The scheme has many nice features such as one secret for multiple services, requiring no certificate, support of user mobility and more importantly, resilience to DoS attacks. We developed our scheme by circumventing a serious weakness in the Wu-Hsu scheme, that is the service provider can learn the secret tokens of the users who have requested services from it. Our scheme has similar performance as the Wu-Hsu scheme, in terms of both computation and communication. While we discuss the proposed protocol in the context of health care system providing a certain service, the protocol can also be a choice for the underlying authentication mechanisms in the systems we shall study in later chapters, as long as user privacy in the login process is a concern. Moreover, the protocol is essentially a generic authentication technique independent of health care systems, but its introduction to health care setting represents a novel use.

CHAPTER 5

Smart Card Enabled Privacy-preserving Medication Prescription

The anonymous remote login scheme we presented in Chapter 4, while considering applications beyond organizational boundaries, deals with relatively simple scenarios in terms of participants, relationship between participants, and system objectives. In this chapter, we study a more complex, inter-organizational process, i.e., medication prescription. Medication prescription is a fundamental routine service in health care systems, involving multiple parties, and individual privacy has distinct implications with respect to different parties. Our focus in this chapter is to clarify and address these privacy concerns. We stress that since this system involves interactions between organizations, the techniques discussed in Chapter 3 and Chapter 4 can be adapted for establishing trust relationship between organizations before they execute the actual task. But we deliberately neglect this as it is orthogonal to the issues we study here.

5.1 Introduction

Within the overall context of health data protection, individual privacy involved in medication prescription needs special treatment. First, the involvement of diverse parties, especially non-medical parties in the process of medication prescription complicates the protection of prescription data. Second, both patients and doctors have privacy stakes, and their privacy concerns should be equally addressed. Third, medication prescription should not be processed in a truly anonymous manner, because (1) certain involved parties need to conduct useful research on the basis of aggregation of prescription data that should be linkable with respect to either the patients or the doctors; (2) prescription data has to be identifiable in some extreme circumstances, e.g., under the court order for inspection or to assign liability. We next give a detailed discussion on these issues.

5.1.1 Privacy in Medication Prescription

As we have made it clear, electronic medical records (EMRs) are gradually substituting traditional paper based medical records in health care systems, providing more efficient and timely information exchange and collaboration among various health care organizations, as well as external business associates. Besides the direct impacts on the quality and efficiency of care provision, the wide use of EMRs eases medical research. For example, researchers in care organizations or research institutes often conduct research on the basis of inspection of clinical data to find or evaluate new therapies; managed care, insurance companies and other providers frequently engage in extensive research on the cost effectiveness of certain medical treatments and practices, by the analysis of health data. While these research are important and beneficial, they pose a potential threat to the individual privacy involved in the underlying health data. From a technical point of view, it seems enough to de-identify the health data prior to use for statistical processing (as we shall see in next chapter, simply de-identifying health data in general does not suffice). However, there are frequent cases in which patients benefit from being traceable by the research, such as in the assessment of treatment safety [126]. In terms of medication prescription, more care is needed to deal with prescription data, as it will be clear shortly.

Ensuring individual privacy contained in medication prescription data is quite relevant in the overall context of health data security [8], primarily due to the fact that prescription data are an indication of a patient's health status and history. In other words, it is by no means very hard to deduce one's health condition by inspecting her prescription data. In this sense, there is little differences between medication prescription data and other kinds of medical records in terms of privacy concern from the viewpoint of patients. Furthermore, doctors also have a privacy stake in medication prescription data since a doctor's prescription habit or pattern is reflected in the data. Such information could be exploited for many purposes. For example a hospital, based on the comparison of doctors' prescription patterns, may issue guidelines on the prescription of certain medicines, and doctors are required to follow; those failing to comply with would be treated negatively. As an another example, medicine companies may take advantage of doctors' prescription information for marketing purposes, tempting doctors to prescribe their medicines [15]. The General Practice Research Database [79] maintained in U.K. serves, among others, exactly this purpose. Patients' information regarding their medication purchasing can be clearly used for a similar purpose by medicine companies.

The process of medication prescription is a little peculiar in the sense that it involves external business associates such as pharmacies and insurers, other than medical related parties. The involvement of multiple parties would inevitably cause multiple points of vulnerabilities in terms of privacy assurance. Moreover, while it is reasonable to presume medical professionals would be bound by ethical obligation and professional faith in ensuring privacy of prescription data, it seems baseless to assume the same for non-medical parties such as pharmacies and insurance companies. Worse yet, legal regulations do not suffice in stopping these organizations from leaking prescription data while they are being used for, e.g., the aforementioned cost-effectiveness research. Take U.S. for instance, there is no federal law on the protection of medical records kept by pharmacies; while on the contrary, pharmacies benefit financially from selling prescription data: over 99% of prescription claims are collected and processed by PBMs (Pharmacy Benefits Management Systems) [81].

On the other hand, protection of individual privacy should not result in an anonymous medication prescription process. If prescription pads were issued in a truly anonymous manner, a wide spread of drug abuses can occur. There was already a thriving black market on prescription medicines [120]. More importantly, laws and regulations require pharmacies to maintain records that can be identified for possible inspection and preventing drug interactions. For example, section 164.512(2)(d) of HIPAA regulates that disclosure of protected health information including audits may be made to health oversight agencies for authorized oversight activities. In addition, some current beneficial research based on prescription data would be rendered impossible once truly anonymous medication prescription is applied. As a consequence, it is desirable that prescription data (1) should be kept *anonymous* in general, but allowed for feasible *anonymity revocation*; (2) provide *linkability* to some parties so as to enable research on the basis of data aggregation. We stress that linkability of prescription data is also conducive to prevention of fraud by patients and doctors in some cases. Considering these, we conclude that (1) prescription data of a patient should be identifiable to the insurer for billing purposes, anonymous yet linkable to the pharmacies (or PBMs) for enabling research or prevention of fraud, and patient anonymity should be revocable under law provision; (2) anonymity of a doctor should be similarly revocable, and prescription from the same doctor should be anonymous (and unlinkable) to the health care organization as well as the pharmacy, but linkable to the insurer for fraud prevention. We shall shortly formalize these points as security requirements upon our medication prescription system.

5.1.2 Use of Smart Card

Easy and instant access to electronically managed health data and insurance information is now a key factor determining the efficiency and quality of care provision. However, the involvement of diverse parties in care process, together with the continuously increased mobility of patients, makes it practically hard to maintain such information in an unified and globally available manner. To be more specific, (1) care provision in general involves a number of parties such as hospitals, clinics, GPs (General Practitioners), and external business associates including insurance companies, billing agencies, pharmacies and so on, resulting in the heterogeneity of information infrastructures and business patterns; (2) mobility of patients comes from the facts that people on frequent trips may need to visit doctors in different cities or even countries; and some patients may need to seek appropriate medical treatment beyond local facilities. It is thus clear that it is hard to achieve the goal of "data availability at the point of care" with the current model of statically maintained information repositories. This difficulty can be resolved to some extent by *smart cards* containing latest personal medical and insurance information, carried by the patients themselves [55, 117].

Medication prescription is among the health care processes that frequently make references to patients' health data and insurance information. In particular, before issuing a prescription, a doctor needs to inspect a patient's medical records, complementing his diagnosis process as well as checking for possible allergies and drug interactions pertaining to the patient; insurance information is consulted to determine whether the intended drugs are indeed covered by the patient's health plan. It is apparent that the introduction of smart card as a *portable* personal information repository would significantly simplify the process of medication prescription, enabling doctors to bypass several bureaucratic and time consuming procedures if otherwise information is retrieved from central databases. Moreover, doctors would be relieved completely from the inconvenience and annoyance caused by occasional blockage of network communication.

In addition to being a data storage device, smart card is capable of moderate computation. We take advantage of this to entitle smart card the digital signature signing capability to sign the electronic prescription pads, asserting the patient's authorization to the prescription before collecting the prescribed medicine. This proof of authorization will be used by the pharmacy to collect payment from the patient's health plan account administrated by the corresponding insurance organization.

Besides the flexibility and convenience in accessing personal health and insurance data, the adoption of smart card in health care systems has many other advantages: the authenticity of the patients is automatically ensured by holding the cards (note that

HIPAA endorses use of smart card for user identification), so that many processes would be automated and sped up, e.g., hospital admissions; with free access to the emergency data stored in the smart card, emergency treatment would be instant; to name a few.

5.1.3 Delegation of Signing in Medication Prescription

Another important observation in medication prescription is that in practice, it is often that other people, instead of patients themselves, collect the prescribed medicine. They may be guardians, relatives, or friends who accompany and take care of the patients to visit doctors. Passing smart card to others for signing prescription pads would definitely increase the likelihood of disclosure of sensitive medical and insurance information stored in the cards, although smart card offers the flexibility to be carried by other people than the card owners. From a technical point of view, it is obviously desirable to root out such a possibility of information disclosure in a practical medication prescription system. Our solution to this problem is to implement *delegation of signing* that enables patients to delegate their prescription signing capabilities to e.g., their guardians, relatives or friends. As a result, the people who have accepted the delegation of signing from a patient can use their own smart cards to sign prescription pads and collect medicine on behalf of the patient. Clearly, delegation of signing avoids the passing of patients' smart cards to the people who actually sign prescription pads, which guarantees that smart card is of total personal use. Delegation of signing is an important functionality in medication prescription, especially for disabled patients. We implement delegation of signing by a proxy signature scheme proposed in the next section.

5.2 A Building Block: Strong Proxy Signature

We have seen the need for delegation of signing in medication prescription, a delegation of prescription signing capabilities. To implement delegation of signing, in this section we propose a strong proxy signature scheme based on the Schnorr signature scheme [159]. And it is straightforward to extend it to other DLP-like signature schemes.

5.2.1 Background and Related Work on Proxy Signature

Participants in proxy signature are *original signers* (delegators) and proxy signers (delegatees), and they work as follows: an original signer delegates her/his signing capability to a designated *proxy signer*, so that the proxy signer is authorized to issue *proxy signatures* on behalf of the original signer. References [139, 181] are among the earliest work on the idea of proxy signature, and the concept was later systematically studied in [135], specifying three types of delegations, i.e., *full delegation*, *partial delegation* and *delegation by warrant*. In full delegation, the original signer simply gives her private signing key to the proxy signer. This kind of delegation seems to have little practical significance as the original signer loses complete control of her/his singing capability. In partial delegation, a proxy signing key pair is generated from the original signer's private key, and the newly generated private key is delivered to the proxy signer through a secret channel. And as the name implies, a delegation by warrant capitalizes on a policy warrant to certify that the proxy signer is trusted. To satisfy the varying requirements of practical applications, combination of the last two types of delegation seems practical and viable. The scheme we propose actually depends on this combination. The schemes proposed in [135] did not offer non-repudiation to the proxy signer since both the original signer and the proxy signer know the proxy signing key. The work in [143] suffered from the same problem. To overcome this drawback, Zhang [191] proposed a non-repudiable proxy signature scheme, which however was found not secure [119]. Reference [121] first introduced the concept of strong proxy signature where a proxy signature issued by the proxy signer represents both the original signer's signature and the proxy signer's signature. Non-repudiation with respect to both the original signer and the proxy signer is thus achieved in strong proxy signature. An earlier scheme in [110] based on the Schnorr signature was in fact a strong proxy signature, whereas it did not reflect the asymmetry of roles that the original signer and the proxy signer take. The strong proxy scheme presented in [121] together with the variant designed for mobile agent environment offer asymmetry of roles, but they are found subject to forgery attacks by the original signer [165]. We develop our scheme by enhancing the scheme in [121] with the robustness to forgery attacks by the original signer. Our enhancement also makes the proxy signer a designated entity, rather than the originally non-designated entity for mobile agents.

To summarize, a strong proxy signature scheme should satisfy the following security requirements:

- **Strong unforgeability**: No one (including the original signer) rather than the designated proxy signer can generate a valid proxy signature.
- **Verifiability**: Anyone can verify the signatures based on the publicly available parameters.
- **Strong identifiability**: A proxy signer's identity can be determined from the proxy signature it generates.
- **Strong undeniability**: The proxy signer cannot repudiate the signatures it gen-

erated.

- **Prevention of misuse**: The proxy signing key pair should not be used for purposes other than the designated ones.

These properties make strong proxy signature an ideal tool for implementing delegation of signing in our system.

5.2.2 A Strong Proxy Signature Scheme

We start by listing notations that are used in this section.

O, PxS, V	the original signer, the proxy signer and the verifier, respectively.
p, q	two large primes such that $q\|(p-1)$.
g	an element of order q in Z_p^*.
(x_o, y_o), (x_{pxs}, y_{pxs})	respective key pairs of O and PxS for the Schnorr signature scheme, with $y_o = g^{x_o} \pmod p$ and $y_{pxs} = g^{x_{pxs}} \pmod p$.
$\sigma(m)$	a digital signature of a message m signed by the Schnorr signature scheme.
$veri(.)$	the verification algorithm of the Schnorr signature scheme.
w_d	a delegation warrant.

Table 5.1: Notations for Proxy Signature Scheme

We are now ready to present our strong proxy signature scheme, which works in the following three procedures.

1. *Delegation*

In the delegation phase, the original signer O chooses $k_o \in_R Z_q^*$, and computes and sends $r_o = g^{k_o} \pmod p$ to the proxy signer PxS. Upon reception of the message, PxS selects $k_{pxs} \in_R Z_q^*$, and in turn computes $r_{pxs} = g^{k_{pxs}} \pmod p$ and $r = x_{pxs}h(y_o||r_o, r_{pxs}) + k_{pxs} \pmod q$. Afterwards, PxS sends (r_{pxs}, r) to O. Upon receiving the message, O computes $s_o = x_o h(w_d, r) + k_o \pmod q$, where w_d is the delegation warrant stating the purposes of delegation, valid period of delegation, etc. O then sends (w_d, s_o) to PxS in a secret channel. PxS checks and accepts as long as $g^{s_o} = y_o^{h(w_d,\, r)} r_o \pmod p$ holds, in which case PxS computes the private proxy signing key as $x_s = s_o + x_{pxs} \pmod q$. As a result, the proxy signing key pair is (x_s, y_s), where $y_s = g^{x_s} \pmod p$ is the public proxy signing key. Note that w_d, r_o, r, and r_{pxs} are public parameters.

2. *Signing*

To issue a proxy signature on a message m on behalf the original signer O, PxS simply computes a Schnorr signature $(\sigma(m))$ using the private proxy signing key x_s, and publishes $(m, \sigma(m))$ as the actual proxy signature.

3. *Verification*

Signature verification is accomplished by checking the following two tests: $g^r \overset{?}{=} y_{pxs}^{h(y_o || r_o,\, r_{pxs})} r_{pxs} \pmod p$ and $veri(m,\ \sigma(m),\ y_s) \overset{?}{=} true$. A proxy signature on m is valid only when both equations hold.

Our scheme has an important feature that both consents from O and PxS are indicated explicitly in the proxy signatures. To see this, r is actually a signature from PxS and s_o is a signature from O. For this reason, the delegation warrant w_d can be simplified. Moreover, recall that r is a signature from PxS on $r_o || y_o$, so it is also a countermeasure against the forgery attacks by the original signer O as suggested in [165], other than demonstrating PxS's acceptance of the delegation.

5.2.3 Security Analysis

We next discuss security of the scheme. As our scheme is an enhancement to circumvent forgery attacks by the original signers as in [121], we begin with the following theorem.

Theorem 5.1 *The proposed strong proxy signature scheme is secure against the original signer's forgery attack.*

Proof: Intuitively, the forgery attacks by the original signer takes advantage of the fact that O is allowed to change r_o by substituting it with $r'_o = y_{pxs}^{-1} \pmod{p}$ (see [165] for detail). In our scheme, however, r_o (together with y_o) is signed by PxS so as to produce r. Since unable to forge PxS's signature, O thus cannot forge r. This avoids the forgery attacks by O. □

We proceed to see our scheme satisfies the security requirements upon strong proxy signature schemes.

Theorem 5.2 *The proposed strong proxy signature scheme fulfills all the security requirements listed above.*

Proof sketch:

(1) Strong Unforgeability: From Theorem 5.1, O cannot forge valid proxy signatures. For outsiders, the private proxy signing key contains PxS's private signing key, therefore only PxS can generate valid proxy signatures.

(2) Verifiability: (r_o, s_o) demonstrates the consent of O on the delegation; (r_{pxs}, r) shows PxS's acceptance of the delegation; verifiability of the signed message is obviously based on the underlying Schnorr signature scheme.

(3) Strong Identifiability: The inclusion of PxS's public key y_{pxs} in the public proxy signing key y_s, implies that PxS is identifiable.

(4) Strong Undeniability: The proxy signer cannot repudiate his signatures because only he can compute the private proxy signing key x_s used in the signatures.

(5) Prevention of Misuse: Expiry date of the proxy signing key can be readily checked against the validity of the keys held by O and PxS, from which the proxy signing key is derived. w_d serves practically to prevent abuses of the proxy signing key. In the context of our medication prescription system, proxy signing keys are intended for the mere use of signing prescription pads. □

5.2.4 A Discussion

An alternative way for generating the proxy signing key is simply that the proxy signer chooses a key pair as the proxy signing key pair and the original signer certifies it using her/his signing key by issuing a digital certificate, and the certificate states the delegation policy. As a matter of fact, there exists a controversy on the practical significance of proxy signature primitives since they do not demonstrate convincing efficiency advantages over this alternative method. Indeed, our proposed scheme faces the same problem. However, one thing is clear regarding our scheme that both the original signer and the proxy signer are explicit from a valid proxy signature itself, together with the public proxy signing key. This as we shall see shortly, is quite critical to make prescription data linkable with respect to patients. Furthermore, since a private proxy signing key contains the private signing key of the proxy signer, the proxy signer thus cannot afford to transfer the proxy signing key pair to others.

5.3 A Privacy Preserving Medication Prescription System

In this section, we elaborate on our construction of a medication prescription system, where individual privacy of patients and doctors is appropriately protected with respect to different parties. Smart card is exploited as a data repository containing latest personal health and insurance information, as well as a computational device for signing prescription pads (by a regular digital signature or our proposed proxy signature). The proxy signature proposed in previous section is used to implement delegation of signing, allowing patients to delegate their prescription signing capabilities to other people.

5.3.1 Basic Idea

Let us look at the (electronic) medication prescription process in real world. A patient visits her doctor, and on the basis of the diagnosis, the doctor will prepare a prescription pad. To that end, the doctor normally connects to the central medical record database for checking allergies and possible harmful drug interactions or medical history concerning the patient. In addition, the doctor may enquiry an information system maintained by the patient's insurer to determine whether certain intended drugs are covered by the patient's health plan. Upon completion of medicine selection, the doctor signs the prescription pad, which would serve as an evidence that the doctor vouches for the safe use of the medicines. The prescription pad is then directed to the pharmacy and added to the patient's medical records. The patient later goes to the pharmacy, and the prescription pad is retrieved. The pharmacy collects enough evidence in filing the prescription to meet the requirements of law regulations. Then the pharmacy charges the insurer (or the patient) for the prescription upon the patient's authorization (signed by the patient) and delivers the prescribed medicines to the patient. The prescription pad may then be forward to the PBMs for statistical research. Our construction will basically follow this processes, but taking protection of individual privacy into account. We next clarify several aspects of our system.

Smart card is useful or even critical to the above process in several places. First, smart card can serve as a portable data repository, containing latest personal health and insurance information. As a result, it is no longer necessary for the doctor to retrieve information from the central databases maintained by the health care organization and the insurance company. This is of a particular advantage when the hospital a patient visits is not her registered care provider. For example, the patient seeks treatment at a different city or at a foreign country. Second, smart card is an especially ideal device for hosting the private signing key, and signing electronically the prescription pad when the patient goes to the pharmacy to collect medicine. Similarly, smart card is also used to host the proxy signing keys for the purpose of implementing delegation of signing, yet another major characteristic of our system. Under our proposed strong proxy signature scheme, to delegate her prescription signing capability, a user (the original signer) negotiates a proxy signing key with the intended person (the proxy signer) who then stores the key in her/his own smart card. A user can be both the original signer who delegates her prescription signing capability to other people, and the proxy signer who accepts prescription signing capability from other people. The accommodation of delegation of signing makes our system more realistic.

Recall that a central objective of our construction is to protect individual privacy of patients as well as doctors in medication prescription, and such a protection should still support useful research on the basis of data aggregation. To that end, we in general adopt the following methods. (1) For patients, each patient applies for a *pseudonym* from her insurer, and the insure links the pseudonym with the patient's real identity. This is reasonable since it is the insurer that pays the prescriptions at the absolute discretion of individual patients. Consequently, patients engage in medication prescription in the name of pseudonyms, thereby gaining anonymity. Transactions under the same pseudonym apparently offers linkability. Revocation of anonymity can be done by the insurer when necessary. (2) For doctors' part, each doctor joins a group established by the care organization. We exploit the security manager of the organization as a *group manager* overseeing the group (for clarity reasons, we call it group manager instead of security manager in the sequel). The group manager holds a key pair for the group, and whenever a doctor issues a prescription, the group manager signs a "group signature" in the name of the group, so anonymity of doctors is achieved. Given a signed prescription pad, only the group manager is able to identify the doctor who issued it. We assume the group manager is independent of the care organization in the sense that the group manager would not do anything in favor of the care organization, e.g., help the organization to link a specific doctor's prescription data. We point out that the functionality of the above "group signature" can be achieved by an off-the-shelf group signature scheme such as [1, 47]. However, virtually all existing group signature schemes are not effective in revocation of group members, thereby insufficient for a dynamic group. For this reason, we choose to let the group manager sign using a regular digital signature scheme on behalf of the doctors. To differentiate doctors while keeping their privacy, the group manger issues each doctor a pseudonym, which serves to hide the real identity of the doctor.

However, it may be argued that the group manager issuing "group signatures" online for every doctor might become a bottleneck, affecting overall performance of the system. Actually, there exist two methods for the group manager to compute "group signatures" in our system, as shown in Figure 5.1. In particular, in case (a), a doctor directly passes the prescription pad m to the group manager for signing, then the group manager signs m and sends the signed pad $\sigma(m)$ to the pharmacy; in case (b), the doctor first delivers m to the pharmacy which later relays m to the group manager for signing. The latter actually offers the flexibility that the prescription can be signed at any time before the patient collects the medicine, alleviating to some extent the situation that the group manager would becomes a system bottleneck. We therefore implement the

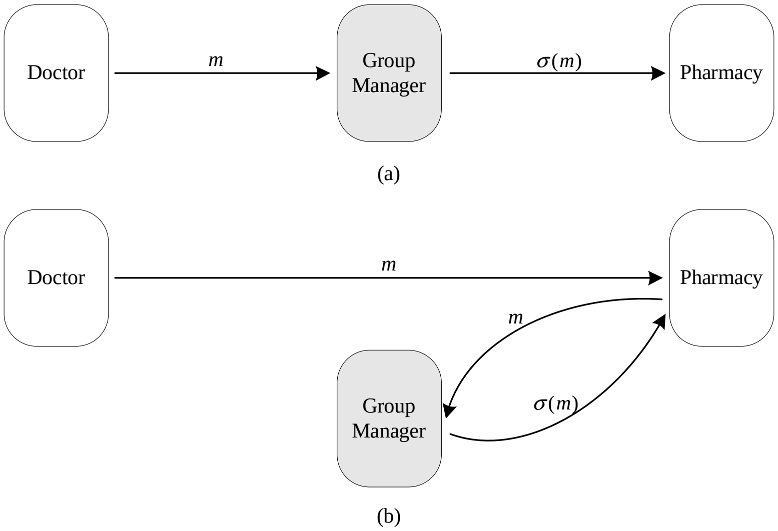


Figure 5.1: Two Modes of Group Signature

latter method in our construction.

As we made it clear, patients (as well as doctors) would rely on pseudonyms to achieve anonymity. However, it is noted that long term linkable pseudonyms would risk the patients being identified. To address this problem, we accommodate the flexibility for readily renewing pseudonyms. In particular, the prescription signing key of a patient is rendered short term. That is, the signing key is certified to be valid within a short period of time, e.g., half a year; or once the patient feels her privacy is at risk, she is able to revoke her pseudonym and the associated signing key (in which case, the signing key is announced in a public CRL (Certificate Revocation List), and then applies for a new pseudonym and a new signing key). The same applies to her proxy signing keys. Under the strong proxy signature scheme in last section, a proxy signing key is derived from the signing keys of both the original signer and the proxy signer. Naturally, revocation of either party's signing key will result in the revocation of the proxy signing key. As a signing key is rendered short term and certified under a pseudonym, it apparently does not suffice for identification purposes in some cases. We then employ a long term key, *master key*, to associate with the real identity of a patient. The master key is intended for user identification and authentication under the real identity of a patient, and may be used beyond the context of medication prescription. As a result, there are three kinds of keys in a patient's smart card, that is, the master key (long term), the signing key for prescription signing (short term), and proxy signing keys (short term) if the patient has accepted delegation of signing from other people.

Based on these discussions, we are ready to formally define the parties involved in our medication prescription system.

5.3.1.1 Definition of Entities

We list in the following the main entities involved in our medication prescription system.

- **Patients**: A patient P is the entity to whom a prescription is issued. P holds a personal smart card hosting her latest health and insurance data. For medicine prescription, P needs to show insurance information pertaining to her health plan (P needs to enrol in a health plan). P also has a signing key in the smart card. To collect the prescribed medicine, P is required to sign the prescription pad using the smart card, to show her consent on the prescription. This authorization will be recognized by the insurer to pay the prescription.
- **Proxy Signers**: A proxy signer PxS accepts delegation of signing from one or several patients. A proxy signer herself may be a patient. As a proxy signer of a patient P, PxS has established a proxy signing key with P and stores the proxy signing key in her own smart card. PxS may be required to sign the prescription pad and collects the prescribed medicine on the behalf of P.
- **Doctors**: A doctor DR is the entity that issues prescriptions. For issuing a prescription, DR signs the prescription pad to claim his assurance of the prescribed medicine benefiting the patient from medical perspective. The signature can be potentially used as a non-repudiable evidence to assign liability if the prescribed medicine caused disputes. To achieve individual privacy, doctors need to join in a group, e.g, established by the care organization they belong to.
- **Insurers**: An insurer I is the party providing health benefits plan to patients, thereby paying the prescriptions. I may need to engage in certain statistical

research. In our system, I issues pseudonyms to the patients who enroled in a health plan, certifies the patients' public prescription signing keys, and revokes anonymity of the patients when necessary. In addition, we designate I to be responsible for detection of fraud committed by doctors.

- **Pharmacies**: A pharmacy PH involves filing prescriptions. In filing a prescription, PH must collect sufficient evidences, including signatures from both DR and P, and collects payment from I and delivers the medicine to P. PH may also engages in statistical research for better medicine provision.
- **Group Manager**: The group manager GM is actually the security manager of the care organization, managing privacy issues of doctors. GM signs prescription pads for doctors who are in the group he manages. GM is responsible for revoking anonymity of doctors when required.

There are other entities involved in our medication prescription system, such as certification authorities that issue public key certificates to related entities, and law enforcement agencies overseeing medication prescription. However, their roles are straightforward, and we do not explicitly discuss them.

We next specify privacy requirements upon medication prescription systems, based on earlier discussions on individual privacy in medication prescription.

5.3.1.2 Privacy Requirements

1. **User anonymity** Actual identities of patients and doctors are hidden by means of pseudonyms. Anonymity, however, can be revoked by the corresponding designated trusted parties. In particular, patient anonymity can be revoked by the

insure I, and doctor anonymity by the group manager GM.

2. **Linkability of patients** Under the provision of anonymity, different prescriptions to the same patient P are linkable to the pharmacy PH. This is essential to enable research by PH.

3. **Linkability of doctors** Under the provision of anonymity, prescriptions issued by the same doctor DR are linkable with respect to the insure I. This is essential for fraud detection by I.

4. **Unlinkability of doctors** Prescriptions by the same doctor DR should be anonymous and unlinkable to the pharmacy PH.

5. **Least data disclosure** Unless absolutely necessary, prescription data are kept confidential. In other words, disclosure of prescription data is based on a need-to-know basis.

Definition 5.1: A medication prescription system is said to be *privacy preserving* if it satisfies the above privacy requirements.

5.3.2 Protocols

In this section, we give our construction of a smart card enabled medication prescription system. Our construction closely follows the real-world medication prescription process described earlier. Our system in general consists of four procedures as outlined in Figure 5.2. Typically, (1) in an off-line *system initialization* procedure, patients enrol in a health plan offered by the insurer I, and doctors join a group managed the group manager GM. Each participant establishes and gets the corresponding keys; (2) in the *prescription preparation* phase, a patient P (or together with a proxy signer PxS) visit a doctor DR, where DR diagnoses P and prepares a prescription pad for P. Subsequently, DR directs the prescription pad to a pharmacy PH; (3) PH then initiates a *prescription signing* procedure, where the prescription pad is forward to the group manager GM for signing (in the name of a group); (4) finally, P or PxS goes to the pharmacy PH to collect the prescribed medicines in the *prescription filing* procedure, and PH gets the payment from the insure I.

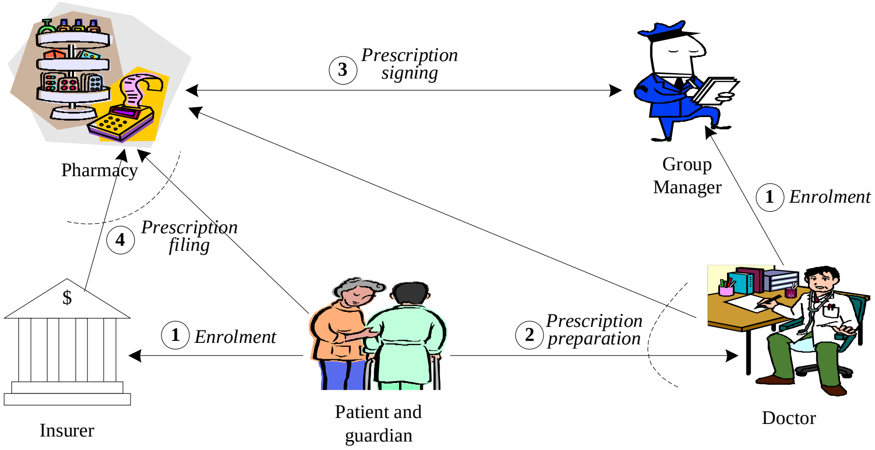


Figure 5.2: A Medication Prescription System.

We next elaborate on these procedures, and the notations that are used in the sequel are listed in Table 5.2.

1. ***System initialization***

In this procedure, each involved entity gets itself prepared for the engagement into the prescription process, including establishing necessary keys and obtaining corresponding certificates.

Suppose the patient P has already established his long term master key (mPK_P, mSK_P) and gotten the certificate under his real identity. P then enrolls in an insurer's health plan. To do this, she establishes her short term signing key (PK_{L_P}, SK_{L_P}), contacts

L_P, L_{DR}	pseudonyms of P and DR, respectively.
TH_i, $i = 0,1,...$	transaction header that minimally contains a transaction ID, inception expiration date, insurance and health plan identifiers.
k_i, $i = 0,1,...$	random session keys.
$E_U(m)$	encryption of m under U's public key by a semantically secure public key cryptosystem.
$\{m\}_k$	symmetric encryption of m under CBC mode with key k.
$EM_{k_1,k_2}(m)$	$\{m, MAC(k_2, m)\}_{k_1}$, where $MAC(k, m)$ is the cryptographic message digest of m with k
$S_U(m)$	digital signature on m by U's private key. We assume cleartext signatures, e.g., $S_U(m) = \sigma(m)\Vert m$, where $\sigma(m)$ is the exact signature of m.
$GS(m)$	"group signature" on m produced by GM.
Rx	a prescription pad.
(mPK_U, mSK_U)	master key pair of entity U; mPK is public key and mSK is private key.
(PK_U, SK_U)	signing key of entity U; PK is public key and SK is private key.
$(pPK_{\succ U}$, $pSK_{\succ U})$	proxy signing key delegated to U from other people.

Table 5.2: Notations for Medication Prescription Protocols.

and directs to the insurer I the public part of the signing key PK_{L_P}. I generates a random pseudonym L_P for P, issues a certificate for the signing key under the pseudonym, finalizes the health plan with P and enters related information together with L_P into a private database for P. Insurance information, L_P and the certificate are delivered to P via a reliable channel, e.g., a registered postal mail. I is also supposed to have a key pair for the asymmetric encryption $E()$. P then negotiates with each proxy signer PxS for delegating her prescription signing capability and helps PxS generate proxy signing keys ($pPK_{\succ PxS}$, $pSK_{\succ PxS}$). P herself may be a proxy signer by accepting others' delegation and generates correspondingly the proxy signing keys ($pPK_{\succ P}$, $pSK_{\succ P}$) that are delegated to her. Finally, public parts of the generated key materials, insurance information obtained from I are added to P's smart card. Note that secret parts of the keys are generated directly inside the smart card. The process is depicted as follows.

$$\begin{array}{lll} (M1) & P \rightarrow I\text{:} & S_p = S_P(\text{Enroll\_Req}, PK_{L_P}) \\ (M2) & I \rightarrow P\text{:} & Cert_{L_P} = S_I(PK_{L_P}, L_P, T), \\ & & E_P(k_1), \{S_I(\text{Insurance\_Info})\}_{k_1} \\ (M3) & P \leftrightarrow PxS\text{:} & \text{establish } (pPK_{\succ PxS}, pSK_{\succ PxS}) \end{array}$$

In $M1$, Enroll_Req is an enrollment request stating which plan to enrol and PK_{L_P} is the public part of the short term prescription signing key. Note that P computes S_p using her master key (mSK_P) to authenticate her real authority to I. In response, I returns to P the certificate $Cert_{L_P}$ under a pseudonym L_P for PK_{L_P} and the insurance information (Insurance_Info) under the enrolled health plan in $M2$. T, included in the certificate, is the expiry date of $Cert_{L_P}$. In order not to be leaked, the signed insurance information is encrypted by a random session key k_1. In $M3$, P exchanges information with a proxy signer *PxS*, establishing the proxy signing key ($pPK_{\succ PxS}$, $pSK_{\succ PxS}$) for *PxS*. P may also set up for himself ($pPK_{\succ P}$, $pSK_{\succ P}$) by accepting delegations from other people. Recall that a proxy signing key is derived from both entities' short term prescription signing keys under the strong proxy signature scheme introduced in last section. So the proxy signing key ($pPK_{\succ PxS}$, $pSK_{\succ PxS}$) is created by (PK_{L_P}, SK_{L_P}) together with (PK_{PxS}, SK_{PxS}), and is valid only when both of them are valid.

A doctor DR joins a group, established by the affiliated care organization, where DR is entitled and certified the capability in issuing prescriptions. The group manager GM of the group will be the actual entity that commits “group signatures” on behalf of the group members. GM issues DR a random pseudonym L_{DR} and certifies DR's signing key (PK_{DR}, SK_{DR}) under the real identity of DR.

To issue "group signatures" for the group members, GM chooses a signing key (note that this signing key is a key pair for a regular digital signature) and obtains the certificate from related certificate authority CA. GM also chooses a key pair for the asymmetric encryption $E(.)$ and obtains the certificate from the corresponding CA.

The pharmacy PH prepares a key pair for the asymmetric encryption $E(.)$ and obtains the certificate from the corresponding CA.

2. ***Prescription preparation***

Patient P visits doctor DR, and presents his personal smart card and signs a random message on the fly to DR, proving his successful enrollment in a particular health plan. The diagnosis process by DR may be complemented by the health data stored in the smart card. Upon completing the diagnosis, DR prepares the prescription. To that end, DR makes references to the medical data in the smart card for checking drug allergies, drug interactions, and insurance information for determining whether certain drugs are indeed covered by P's health plan. DR then generates an electronic prescription pad including no identities of P and DR. Afterwards, DR delivers the prescription pad together with the information regarding L_P to the pharmacy PH. Note that DR should be anonymous to PH. Finally, DR updates P's smart card by adding to it the particulars of current visit and prescription.

$$
\begin{array}{lll}
(M4) & P \rightarrow DR: & S_{l_p} = S_{L_P}(Tstmp),\ Cert_{L_P} \\
(M5) & DR \rightarrow PH: & E_{PH}(k_2, k_3), \\
 & & e = EM_{k_2,k_3}(TH_0,\ Rx,\ S_{l_p}),\ Cert_{L_P}, \\
 & & Pe = E_{GM}(L_{DR},\ S = S_{DR}(TH_1,\ e))
\end{array}
$$

In particular, in $M4$, P computes a signature S_{l_p} on $Tstmp$, the current timestamp, using his prescription signing key to show his successful enrolment in a health plan offered by insurer I. The prescription is sent by DR to PH in $M5$, where k_2 and k_3 are random session keys for PH to decrypt and check e; TH_0 and TH_1 are transaction headers as defined in Table 5.2; Rx is the prescription pad including a serial number Prescription_Id; Pe is intended only for GM to decrypt, and S is a signature on e under the real identity of DR which serves to tell GM who issues the prescription. $Cert_{L_P}$ included in $M4$ and $M5$ is used to verify S_{l_p}.

3. ***Prescription signing***

The pharmacy PH transfers the prescription to the group manager GM for signing. To minimize the likelihood of leaking prescription information, it makes sense to hide the exact prescription content from GM. This however will not cause trouble because GM is in charge of anonymity revocation of doctors, so is able to keep the scrambled message traceable; this would also prevent GM from otherwise substituting certain drugs for discriminative purposes against P. Therefore, in our system, GM issues a "group signature" to the encrypted prescription. GM includes in the "group signature" a linkable token in an attempt for insurer I to link doctors' data. GM then returns the signed prescription to PH. The process is illustrated by the following steps:

$$(M6)\ \ PH \rightarrow GM\text{:}\ \ Pe,\ Cert_{L_P}$$

$$\begin{aligned}(M7)\ \ GM \rightarrow PH\text{:}\ \ & Gs = GS(TH_2,\ e,\ \{DR,\ S\}_{k_4},\\ & E_{GM}(k_4),\ \tilde{e} = E_I(L_{DR}))\end{aligned}$$

In $M6$, PH relays Pe received in $M5$ to GM. GM then decrypts to get L_{DR} and

S. Since S is a signature (under DR's actual identity) on e, GM verifies e. From L_{DR}, GM retrieves from his database the real identity corresponding to L_{DR}, and checks against the one indicated by S. In $M7$, GM returns to PH the "group signature" on e, where TH_2 is a transaction header, k_4 is a random session key and is encrypted by GM's public key, so $\{DR,\, S\}_{k_4}$ can be opened only by GM; $\tilde{e}$ is the ciphertext created by insurer I's public key, thereby openable only by I, and $\tilde{e}$ is intended for I to link doctors' prescription data. Since PH keeps an original copy of e, he can detect GM's possible modification of e by comparing the returned signed e with the original copy. Apparently, PH has also no chance to substitute drugs in the prescription.

4. ***Prescription filing***

To collect the prescribed medicines, the patient P or the proxy signer PxS goes to the pharmacy PH, where P or PxS signs the prescription pad using her own smart card. Signatures of both P and DR are the evidences that must be collected by PH in compliance with law regulations for legal sale of medicines. PH gets the electronic payment from the insurer I by providing I the prescription record, and delivers the medicine to P or PxS. The following steps outline the process.

$(M8)$	$PH \rightarrow P$:	$k_2,\, k_3,\, Gs$
$(M8')$	$PH \rightarrow PxS$:	$k_2,\, k_3,\, Gs$
$(M9)$	$P \rightarrow PH$:	$\tilde{S} = S_{L_P}($ Prescription_Id, $S_{l_p})$
$(M9')$	$PxS \rightarrow PH$:	$\tilde{S} = S_{PxS}($ Prescription_Id, $S_{l_p})$
$(M10)$	$PH \rightarrow I$:	$E_I(k_2, k_3),\, Gs,\, \tilde{S},\, Cert_{L_P}$
$(M11)$	$I \rightarrow PH$:	Electronic Payment, $S_i = S_I($Prescription_Id, $S_{l_p})$

Specifically, prior to signing, P or PxS must verify the prescription. To that end,

PH submits Gs to P's (or PxS's) smart card in $M8$ (or $M8'$), where k_2 and k_3 are the same session keys as in $M5$ for decrypting e included in Gs. Note that we assume the submission channel from PH's workstation to the smart card is secure, so k_2 and k_3 are sent in clear. Upon confirmation, P or PxS signs the prescription in $M9$ or $M9'$. The Prescription_Id, together with S_{l_p} obtained from e, uniquely identifies a prescription. To collect payment, PH forwards the signed prescription Gs, signature $\tilde{S}$ and the encrypted session keys k_2, k_3 to the insurer *I.* Upon validating the prescription, I pays the bill and returns a signature S_i to *PH.* At this point, a successful prescription session completes and PH may pass the prescription data to a PBM for statistical research. Gs, $\tilde{S}$ and S_i are a set of complete evidences of a prescription to be collected by *PH.* Note that we have avoided the prescription to be signed in a recursive fashion, i.e., one entity signs upon another entity's signature. Verifying such a recursively signed message must proceed in a sequential manner. Instead, Gs, $\tilde{S}$ and S_i can be verified independently and in parallel.

5.3.3 Security Discussions

In this section, we discuss how the above protocols meet the earlier privacy requirements.

Theorem 5.3 *The proposed E-prescription system is privacy preserving, satisfying the privacy requirements.*

Proof sketch:

(1) **User anonymity**. *User anonymity requires that actual identities of patient P and doctor DR are appropriately protected, but revocable to the designated entities.* In the above construction, P and DR engage in the process of prescription with respective pseudonyms, with the only exception in the *system initialization* step. In particular, P interacts under its real name with the insurer I to apply for a pseudonym as well as the certificate for the prescription signing key, and to negotiate health plan; DR communicates with the group manager GM to acquire its pseudonym and credential for issuing prescription. Both cases, however, are deemed reasonable considering the fact that I and GM are the designated entities for anonymity revocation of I and DR, respectively. The real identity of DR is also included in messages $M5$, $M6$, $M7$, $M8$ ($M8'$) and $M10$. But notice that in all cases, only GM can decrypt the corresponding ciphertexts to read the identity. Moreover, no identity information of P and DR is incorporated in the prescription pad Rx. Considering these facts, anonymity of both patients and doctors are achieved.

Anonymity revocation of P is clear in the sense that given any signed prescription data under the pseudonym L_P, only the insurer I can map L_P to the real identity of P. As to DR, in $M7$, GM includes $\{DR, S\}_{k_4}$ and $E_{GM}(k_4)$ in Gs, which are readable only to GM and thus anonymous to other entities. This suggests that given a valid prescription data Gs, only GM can tell which doctor exactly issued the prescription.

(2) **Linkability of patients**. *It requires that under the provision of anonymity, prescriptions to the same patient P are linkable to pharmacy PH.* Linkability of patient to PH follows immediately if the prescriptions to P are signed by P himself in $M9$. If the prescriptions to P are signed by a proxy signer PxS in $M9'$, according to a property of our proposed strong proxy scheme, i.e., identities of both the original signer and the proxy signer are explicit in a valid proxy signature, linkability of the patient is also achieved.

(3) **Linkability of doctors**. *It requires that under the provision of anonymity, prescriptions issued by the same doctor DR are linkable wit respect to insure I.* Prescriptions issued by the doctor are signed by group manager GM in $M7$. GM includes $\tilde{e} = E_I(L_{DR})$ in the group signature Gs. Since the insurer I is able to decrypt $\tilde{e}$ using its private key, linkability of doctors to I is thus achieved. $E()$ is a semantically secure public key cryptosystem, by reading $\tilde{e}$ without decryption, no one can do the same linking.

(4) **Unlinkability of doctors**. *It requires that prescriptions issued by the same doctor are anonymous and unlinkable to pharmacy PH.* Anonymity of doctors to PH holds true as we already discussed in the first requirement. It then suffices for us to show that Gs is unlinkable to PH. What included in Gs are TH_2, e, $\{DR, S\}_{k_4}$, $E_{GM}(k_4)$ and $\tilde{e}$: TH_2 is random; e and $\{DR, S\}_{k_4}$ are also random encrypted by random session keys; so is $E_{GM}(k_4)$; and as we just discussed, from $\tilde{e}$ no one including PH can do the same linking as I who can decrypt $\tilde{e}$. Unlinkability of doctors to PH is thus achieved.

5) **Least data disclosure**. *It requires that unless absolutely necessary, prescription data should be kept confidential.* It would be quite hard to precisely define and then prove the implication of least data disclosure in the system. We however mention two salient facts of our system relating to this requirement. First, to protect the information including the prescription data stored in a patient's smart card, the patient delegates her signing capability to other people to avoid her card being carried by others, which otherwise may risk disclosing information in the card. Second, to avoid unnecessarily disclosing information while without affecting the responsibilities it takes, the group manager GM is designed to "blindly" sign prescriptions. □

5.3.4 Revocation of Delegation of Signing

In some cases, a patient may want to revoke the delegated prescription signing capability of a proxy signer. Revocation of delegation of signing in our system can be achieved following a similar way as revocation of public keys in a PKI. More specifically, a delegation of signing revocation list (DoSRL) is maintained by e.g., the insurer who issues public key certificates to patients (the original signer and the proxy signer); each item in the DoSRL contains a pair of public keys of the original signer and the proxy signer, suggesting that the delegation relationship between the two keys was revoked; as such, the validity of a strong proxy signature must first be checked against the DoSRL to see whether the involved keys have been published in the DoSRL: if the keys are in the DoSRL, the proxy signature is definitely invalid; otherwise, it continues the regular Verification function of the strong proxy signature scheme. It should be noted that DoSRL works exactly as CRL (Certificate Revocation List) in a PKI.

5.4 Smart Card Aspects

Needless to say, security of the smart card is of paramount importance in our system. We consider the smart card as a tamper resistant device that offers significant resistance to physical attacks. The smart card is equipped with a crypto-coprocessor for performing crypto-algorithms. The SLE66CX microcontroller family from Infineon Technologies [16] and the AT90SC microcontroller family [92] from Atmel seem suffice for our system since they perform fast discrete logarithm computations by hardware. There are normally three types of memories constituting the storage system of a smart card, namely *working memory*, *program memory* and *user memory.* Working memory is made up of Random Access Memory (RAM) chips, providing temporary storage for the data exchanged during program execution. Data in working memory will get lost when power is off. Program memory is a kind of nonerasible Read Only Memory (ROM). The operating system and the security module that enforces security mechanisms reside in this area. The content of program memory is entered when the chip was manufactured, and any later attempt to modify it would ruin the card. User memory, taking advantage of EEPROM technology, is programmable in the sense that it can be erased and re-written by electronic means. All personal data used in our system including medical records, insurance information, key materials (master key, signing key and proxy signing keys from other people if any) are stored in this area.

We organize the user memory into distinct sections to accommodate data requiring different maintenance and access control. Note that the allocation of space is theoretical, and the precise structure and the data access control will be implemented in accordance with existing standards [94–96].

- Secret Section

  This section is designed to be written only once and cannot be read from the outside by either physical or logical means [173]. The data in this area are retained throughout the life cycle of a smart card, and can only be read by its own microprocessor. The following data are archived in this section.

  - the card manufacture's PIN.
  - the card holder's long term master key: The master key serves to authenticate the patient's actual identity, e.g., when the patient enrols in a health plan by interacting with the insurer.

- Sensitive Section

  This section is similar to the secret section, but allows for occasional updates. The following information is stored here.

  - the card issuer's PIN (CIN): The card issuer in our system may be a patient's primary health care provider organization. CIN serves to protect the application data against unauthorized operations such as erase and write.
  - The card holder's PIN (CHN): The card holder is obviously the patient herself in our system. CHN is used to activate certain functionalities of the smart card, e.g. to review the protected information.

- Working Section

This section can be erased and rewritten, whereas such updates can be accomplished only by designated entities, the card issuer or holder in our case. The information in working section is read protected, write protected, or erase protected through appropriate access control codes (CIN or CHN), depending on the nature of the data. The following data are managed in this section.

  - private part of the card holder's short term prescription signing key: The signing key serves to sign electronically the prescription when the patient collects the medicine in the pharmacy.
  - private part of the short term proxy signing keys delegated to the card holder: The card holder may agree to be the proxy signer of other people in terms of prescription signing. Be this the case, the proxy signing keys are stored in this area.
  - medical information: the medical information set includes coded personal medical records, consultation details and prescription information.
  - insurance information.

- <u>Public Section</u>

  Data in public section can be read free, requiring no protection. The following data are stored in this area.

  - serial number of the card.
  - pseudonym and related information.
  - emergency medical information: such information includes blood type, drug allergies, etc.
  - public keys and their corresponding certificates: These include the delegation

warrants stating delegation policy for the use of the proxy signing keys.

To summarize, we list the data (and sections) managed in the user memory in Table 5.3. Note that the size quantities are not accurate, normally bigger than the actual data sizes.

Data (Section)	Size (bytes)	Reading	Erasing	Writing
Secret section	40	Forbidden	Forbidden	Forbidden
Sensitive section				
CHN	10	Forbidden	CHN	CHN
CIN	10	Forbidden	CIN	CIN
Working section				
signing key	30	Forbidden	CIN	CIN
delegated keys	90 (30×3)	Forbidden	CIN	CIN
medical records	40	CHN	CIN	CIN
consultation info.	1,500 (50×30)	CHN	CIN	CHN
prescription info.	1,200 (120×10)	CHN	CIN	CHN
insurance info.	250	CHN	CIN	CIN
Public section				
pseudonym info.	10	Free	CIN	CIN
emerg. med. info.	20	Free	CIN	CIN
pub. sig. key	450	Free	CIN	CIN
pub. prox. keys	1,350 (450×3)	Free	CIN	CIN

Table 5.3: Data Management in Smart Card

We clarify some particulars presented in the table.

- By *Reading Forbidden*, the data can only be read through the microprocessor of the smart card.
- The design of the data structure for medical record is merely indicative instead of descriptive. In other words, we *code* the medical record using a well-structured template. As a result, most of the fields accept binary values "YES" or "NO". Reference [117] provided an example of such a structured template. For example, if a patient has "*Obsessive-compulsive disorder*", the corresponding field will be

"1". Similarly, all fields are filled with either "1" or "0". In this way, the 40-byte space allocated for the patient's medical records can accommodate 320 fields.

- We assume that discrete logarithm based public key cryptosystems (e.g., the Schnorr signature scheme [159]) are used to compute digital signatures and issue key certificates. This makes typically 160-bit private keys, 256-byte public keys, and 148-byte digital signatures. A public key (short term) certificate is simplified to contain minimally the user's name, CA's name, expiry date and a digital signature on them. Other certificates, such as those for proxy signing keys, may contain a simplified version of policy. With these, the length of a public key together with its certificate is expected not to exceed 450 bytes.
- For the master key, as it is for long term use, the public key certificate should be produced in a standard format. For the limited space, we don't include this certificate in the smart card, thereby not providing a verifier the convenience to verify a signature off-line. This however does not degrade the efficiency of our medication prescription system, for the master key is used only once in the initialization phase.
- The area for consultation details and prescription information is writable under the card holder's PIN (CHN). With this, our system offers the flexibility that such information can be added to the smart card under the authorization of the patient. This is significant when the patient visits a doctor in other place than his primary health care organization.
- We allow information regarding the latest 30 consultations and 10 prescriptions being stored in the smart card. Removal of this kind of information is on a "first

in, first out" basis. For the limitation of space, a card holder is permitted to be the proxy signer of at most three people. Therefore, maximally 1,350 (450×3) bytes of space is allocated for proxy signing keys and their certificates.

- The total space to accommodate all the data is estimated to be 5 Kbytes. Therefore, a smart card with 8 Kbytes memory suffices for our system.

As a final note, we point out some existing health card systems for the comparison with ours. The Health Smart Card in Texas [82] serves mainly as a medical data container, and the Health Card in France [140], besides containing health care information, is intended more as a paying means for health services. The Health Professional Card (HPC) [35] has been standardized on European level as CEN prEVN 13729 "health Informatics - Secure User Identification - Strong Authentication using Microprocessor Cards" [46] as well as consistently on the German national level as the HPC Protocol [89]. HPC is more on providing identification services with security functionalities such as digital signature and encryption.

5.5 Concluding Remarks

We have proposed a smart card enabled medication prescription system, with the following features distinguishing it from the system in [8]. First, the introduction of smart card carrying personal health and insurance information greatly simplifies the process of diagnosis and medication prescription, while smart card in [8] is used only for prescription signing. Second, pre-approval for a prescription from the insurer in [8] is no longer deemed necessary in our system, because doctors can get enough insurance information from a patient's smart card to support the prescription preparation procedure. Third, we identified and accommodated the need for patients to delegate their prescription capabilities to other people, e.g., their guardians. This is good to protect privacy of the information stored in personal smart cards, making our system more acceptable in practice. The work in [8] did not consider delegation of signing.

We believe that our proposed system is quite practical considering smart cards have already been deployed in some health care systems, e.g., [35, 82, 140]. Implementation of our construction at the smart card level is our future work.

CHAPTER 6

Privacy and Ownership Preserving of Health Data in Outsourcing

In Chapter 5, we discussed to achieve user privacy (patients and doctors) towards some organizations that are involved in the medication prescription process, while still enabling these organizations to collect useful prescription data for the purpose of conducting research based on these data. The work in this chapter in general continues such kind of study on “achieving user privacy while enabling research”, whereas we consider a different scenario: a health care organization (e.g., a hospital) outsources the health data in its repository to other organizations (e.g., medical research institute) so as to enable research by the receiving organizations. For example, a hospital may need to outsource clinical records in its autonomous databases to a research institute in an attempt to discover a new drug or evaluate a new therapy. A main difference between the scenarios considered in this chapter and in Chapter 5 is that the data to be shared (outsourced) in this chapter are an aggregation of medical records while in Chapter 5 are individual records (a record for every prescription session). This suggests that we need to consider privacy protection at a level beyond individual data items, e.g., some statistic properties of the whole data set should be taken into consideration. Moreover, the outsourcing care organization in this chapter does not have direct business association with the receiving organizations with respect to the data to be outsourced. Consequently, the receiving organizations actually involve “secondary” use of medical records, and outsourcing care organization has more interests to be protected from the receiving organizations.

It is now clear that we are concerned with protection of health data in secondary use, in which case, two important issues have to be addressed: one is the privacy protection for individuals referred to in the outsourced data; the other is copyright protection over the outsourced data. We present a unified framework that seamlessly combines techniques of binning and digital watermarking to attain the dual goals of privacy and copyright protection. Our binning method is built upon an earlier approach of generalization and suppression by allowing a broader concept of generalization. To ensure data usefulness, we propose constraining binning by usage metrics that define maximal allowable information loss, and the metrics can be enforced off-line. The watermarking algorithm we propose watermarks the binned data in a hierarchical manner by leveraging on the very nature of the data. The method is resilient to the *generalization attack* that is specific to the binned data, as well as other attacks intended to destroy the inserted mark. We prove that watermarking could not adversely interfere with binning. We implemented the framework and conducted extensive experiments on the algorithms, and the results show the robustness of the proposed framework.

We remark that the entity in a care organization responsible for outsourcing is the security manager of the organization. However, we do not explicitly discuss how the security manager enforces the protection mechanisms and how the security manager contacts the receiving organization, as this is straightforward and orthogonal to the issues we study here.

6.1 Introduction

Nowadays, effective sharing of health data is essential to foster the collaboration within the health care community and with other parties such as research institutes, managed care, and pharmaceutical companies, so as to enhance the quality and efficacy of care provision. For example, a hospital may need to outsource clinical health records in its autonomous databases to a medical research institute in an attempt to discover a new therapy or evaluate a new treatment. Such need is clearly shown by research trends in the area of health care management and procedures that are increasingly based on extensive analysis of clinical data. And it is well recognized that research of this kind promises many advantages such as improvements in care provision, reduction in institutional costs, enhancement in organizational administration, better treatment alternatives, development of predictive and diagnostic support systems, to name a few. These benefits, however, come at the expense of care organizations outsourcing health data for secondary use.

The dissemination of health data could also be to satisfy legal requirements. As reported by the National Association of Health Data Organization in 1996, 37 states in the United States had legislative mandates to gather personal health information from hospitals for cost-analysis purposes [138].

The direct release of health data invariably violates individual privacy. Data must be thus properly processed before delivery in order to protect privacy of the individuals they refer to. A straightforward method for achieving individual privacy is to de-identify (anonymize) the data, by replacing any explicit identifying information by some random placeholders. For instance, a randomized value may be used to substitute the name or social security number of each patient. This alone, however, does not suffice to guarantee the full anonymity of medical data as pointed out by numerous studies (see for example, [91, 168, 170, 172]). An example often outlined is *re-identification* by linking attributes such as birth date, zip code that are shared by the anonymized health data and some externally collected voting records. This has motivated many more advanced approaches in the literature (see Section 6.2). Of particular interest is the approach of *generalization and suppression* [168,170,172] that represents values by corresponding more general but semantically accordant alternatives.

The sharing of health data also exposes data holders to the threat of data theft. Related to this, yet another important protection requirement regarding outsourced health data arises, that is, how to protect data ownership (copyright). It is quite obvious that health data are an important asset to the data holders (health care organizations) who have collected and compiled the information. Incentives to unauthorized data distribution arise from an increasingly thriving *data industry* where firms such as biotech companies collect, compile, share or sell (bio)medical data for profits. Even though there are laws concerning copyright and ownership rights, we need effective mechanisms to establish and protect the holders' rightful possession of the data. Consequently and naturally, digital watermarking techniques, initially proposed for the protection of multimedia content [41,100], have been recently also applied to relational data. As such, digital watermarking techniques represent a viable solution for the problem of enforcing ownership of health data. However, a main difference of health data with respect to data from different domains is represented by the need of also assuring privacy. It is thus clear that when dealing with outsourced health data, both individual privacy and data ownership must be protected. To meet these dual needs, we propose a framework that integrates techniques of binning and digital watermarking. Under our framework, the health data to be outsourced would undergo two consecutive steps of binning and watermarking, respectively.

To summarize, the main contributions of our work in this chapter include:

1. A unified framework that seamlessly combines binning and digital watermarking for the protection of both individual privacy and data ownership. We give both theoretical and experimental analysis on the "seamless-ness" of the combination.
2. A binning algorithm that enforces the functionality of "binning". The method bins downward, and extends an earlier approach of generalization and suppression by allowing a broader concept of generalization.
3. A hierarchical watermarking scheme that is resilient to various attacks attempting to remove the embedded mark, and especially robust against the newly discovered *generalization attack*. In addition, we propose an elegant solution to the rightful ownership problem concerning watermarking.
4. The adoption of usage metrics for preserving data quality with respect to the intended usage. We define our usage metrics by modelling information loss, and propose an off-line enforcement of usage metrics.
5. Experimental studies of the proposed framework.

Compared to existing approaches, a main innovative aspect of our work is represented by a downward binning process to address the satisfaction of k-anonymity specification, due to the off-line enforcement of usage metrics; our watermarking algorithm is a novel hierarchical scheme that exploits the very nature of the underlying data, which also provides a neat solution to the rightful ownership problem.

We start by providing some background knowledge as well as related techniques in next section.

6.2 Background and Related Techniques

Conceptually, the health data to be outsourced and thus protected can be viewed as a relational table organized into rows and columns. Each row of the table is a record describing an individual/entity, and each column represents a distinct attribute of all individuals/entities. For example, a table for patient information would store patients' name, social security number, zip code, race, birth date, gender, visit date, and so on, as a series of columns, and a row of the table is a record about a particular patient. A column is essentially a semantic domain comprising of a set of possible values. A row is also termed a tuple, which consists of an ordered n-tuple of values, where n is the number of columns. Based on the identifying information they contain, columns are categorized into three types. Some columns explicitly identify individuals (e.g., name, social security number), so they are called identifying columns. Some other columns contain potentially identifying information that could be linked with other data sets to re-identify individuals, even without the presence of identifying information. Such columns are called quasi-identifying columns. Typical examples of quasi-identifying columns include zip code, birth date, gender and so on. The rest of columns contain no identifying information. Of particular interest is the quasi-identifying columns with respect to the protection of individual privacy as the identifying columns can be simply substituted by randomized tokens and columns containing no identifying information in fact need no disposal at all. Therefore, unless explicitly stated, our later discussions are restricted to the quasi-identifying columns.

Basically, two classes of technique are closely related to our work, namely, information disclosure control and database watermarking, respectively.

6.2.1 Information Disclosure Control

Nowadays, person-specific data are collected widely, and then shared for business or legal reasons. An important issue has to be addressed is the control of information disclosure, i.e., the privacy of individuals/entities referred to in the released data must be protected. Information disclosure comes in the form of either the identity of an individual is directly revealed or something about an individual is learnt from the released data. By convention, we call the former *identity disclosure* and the latter *attribute disclosure* [116]. Attribute disclosure in a broad sense can include *inferential disclosure* whereby certain characteristic of the individuals can be inferred by analysis of the released data [61]. In this work we will restrict ourselves to the identity disclosure problem, and we refer interested readers to [184] for in-depth discussions on attribute disclosure.

As stated earlier, simple de-identification (anonymisation) by stripping or replacing explicit identifying information such as names or social security numbers in the underlying data does not suffice to protect the privacy of individuals. Experiences showed that linking the anonymized data with other externally available data sets such as regional voting records and hospital discharge records, still risks re-identifying individuals [91, 133, 168, 170, 172].

One well known approach to identity disclosure control is to transform quasi-identifying columns to entertain k-anonymity constraint (k is a constant), i.e., data are generalized and suppressed in such a way that every record is indistinguishable from at least k-1 other records, so that no search can be narrowed down to a particular individual [91, 168, 170, 172]. The satisfaction to k-anonymity can also be understood as: records containing the same value constitute a bin, and the size of every bin is at least equal to k. By definition, generalization deals with replacing a value with a more general but semantically accordant value, while suppression deals with preventing data releases. Generalization of categorical attributes is based on the fact that the representation of medical data can be normally arranged into a *domain hierarchy tree* (DHT), where the most general description of the data is at the root of the tree while the leaves denote the most specific descriptions. Figure 6.1 shows a DHT on the type of roles: leaf nodes represent all possible particular roles a column may assume, and generality of the description increases with the level along the tree, until the root node that distinguishes no specificity. A generalization proceeds by replacing the values represented by the leaf nodes by their corresponding ancestor nodes at a higher level. For instance, the set of {Neurologist, Gynecologist, Radiologist, Cardiologist} may each be generalized as Doctor, Medical Personnel, or even Hospital Staff in Figure 6.1, depending on the level of privacy it aims to achieve. A valid generalization in [168, 170, 172] requires all its generalization nodes be at the same level in the domain hierarchy tree.

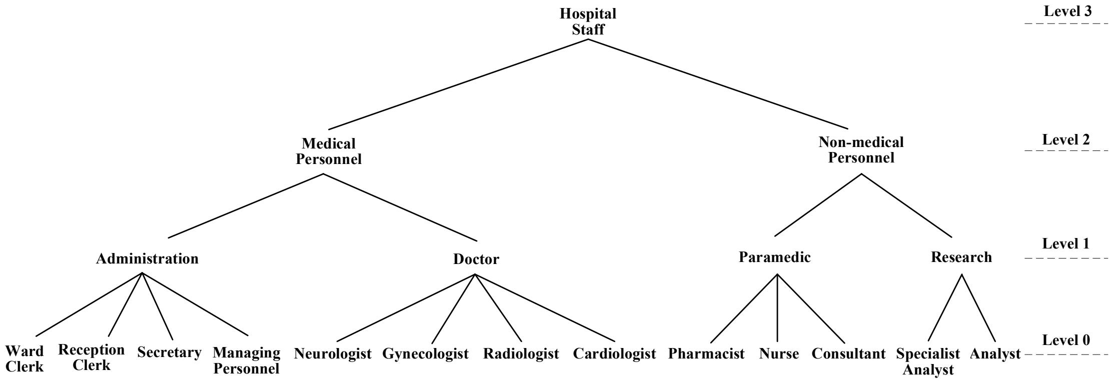


Figure 6.1: A Domain Hierarchy Tree (DHT) for A Column Tepresenting the Types of Roles in A Medical Domain.

Clearly, generalization and suppression result in a loss of specificity, thereby making the re-identification process harder. However, the tradeoff between the level of privacy and the amount of information loss must be carefully evaluated, as too much generalization could possibly render the data useless while slight generalization could not provide adequate protection. [97] suggested associating usage based metrics with the process of meeting k-anonymity. Our framework incorporates the same idea of usage metrics, but we define a different set of metrics, and more importantly, our metrics can be enforced off-line. Metrics in [97] are defined in accordance with the broader notion of generalization allowed therein, which does not require all generalization nodes stay at the same level. The *binning* method in [118] follows a similar broader definition of generalization. Considering the flexibility and finer granularity it offers, our binning algorithm also includes such a broader notion in extending the generalization and suppression in [168, 170, 172]. Moreover, the off-line enforcement of usage metrics enables a downward binning in our context, which has efficiency advantage over binning that proceeds upwards.

Another approach to the identity disclosure problem is to perturb the data by adding noise or swapping values, while at the same time maintaining some statistical properties of the entire data set [72, 114]. Perturbations apparently cause data loss, so it is again vital to determine the right tradeoff between information loss and privacy – a topic which is now under active research [62, 188]. From the discussions so far, we know that identity disclosure control essentially deals with sharing data in such a way that the released data remains useful with respect to its intended usages while safeguarding the privacy of individuals to which the data refer.

6.2.2 Watermarking of Relational Data

Digital watermarking has long been investigated for copyright protection, mainly over multimedia content, e.g., images, audio and video clips [41,100]. There have been recent efforts in watermarking relational databases. Due to the very nature of relational data such as low noise bandwidth, strict definition of semantics, etc., watermarking techniques for databases turned out not to be a direct deployment of techniques for multimedia data. A seminal approach to watermarking relational data is presented in [7]. However, the use of Least Significant Bits (LSB) embedding in the scheme makes it inherently vulnerable, as a simple flipping of LSBs would completely destroy the inserted mark. [155] proposed a method for watermarking numbers that is robust because the mark embedding relies on data distribution rather than on trivial LSB modification. The idea has later been integrated in a framework for watermarking numeric attributes of relational databases [156]. A theoretical investigation on watermarking techniques for databases and XML documents is presented in [5], which attempts to achieve watermarking while preserving a set of parametric queries in a specified language.

Another approach [171] was recently proposed dealing with watermarking categorical attributes in databases. In essence, the data to be watermarked in our context become categorical after binning, so our watermarking also reduces to handling categorical data. Unfortunately, such approach cannot be directly applied to our case because it is susceptible to a kind of *generalization attack* (see Section 6.5).

6.3 Overview of Our Framework

To simultaneously attain the goals of protecting individual privacy and copyright protection regarding outsourced health data, we combine techniques of binning and digital watermarking into a unified framework. As shown in Figure 6.2, the framework comprises two key components, i.e., binning agent and watermarking agent, dedicated to binning and watermarking, respectively. In the framework, the health data to be outsourced would undergo two consecutive steps of transformation. Specifically, the binning agent first bins the data to satisfy k-anonymity specification. Afterwards, the binned data are watermarked by the watermarking agent by inserting within the data a mark, which, upon extraction, asserts provable ownership. The data resulting from these transformations are then expected to adequately protect both privacy and copyright, thereby qualified for outsourcing. Both binning and watermarking are governed by usage metrics in order to preserve data usability. Next, we shall discuss some specific aspects of the framework.

Usage Metrics: Usage metrics define a set of maximal distortions that binning and

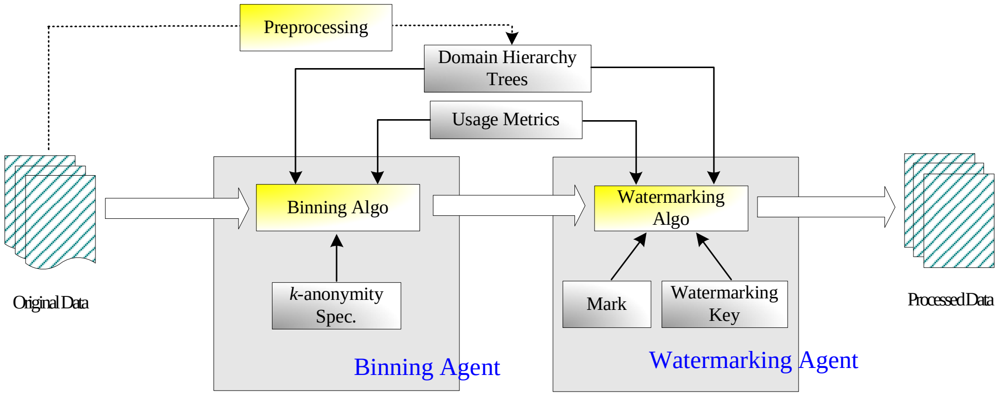


Figure 6.2: Protection Framework for Outsourced Health Data.

watermarking are allowed to introduce with respect to the intended data usage (see Section 6.4). Transformation exceeding the bounds is assumed to render the data useless.

k-anonymity Specification: k-anonymity specification includes the system parameter k, and possibly also the set of quasi-identifying columns to be binned and other relevant constraints pertaining to binning.

Binning Agent: Driven by the binning algorithm, the binning agent attempts to bin the data to satisfy k-anonymity specification while at the same time adhering to the usage metrics. After binning, each bin is guaranteed to contain at least k records, so no specific individual can be identified. The binning algorithm takes as input the original data, the k-anonymity specification, the domain hierarchy trees for each quasi-identifying attribute, and the usage metrics. We suggest a preprocessing step to create the domain hierarchy trees and determine the system parameters.

Watermarking Agent: The watermarking agent continues to process the binned data by embedding an owner-specific mark. The underlying watermarking algorithm exploits a secret watermarking key (may contain several elements), known only to the data owner, to manipulate the process of mark embedding. Without having possession of the secret watermarking key, no one can erase the inserted mark from the data. Watermarking also observes usage metrics, ensuring that it does not corrupt the data in terms of the anticipated usage; the domain hierarchy trees are needed as well for inspection by our watermarking algorithm.

6.4 Binning Algorithm

Our binning algorithm extends the approach of generalization and suppression in [168, 170, 172] by allowing a broader notion of generalization as in [97], which does not require all generalization nodes of a generalization to be necessarily at the same level of the domain hierarchy tree. In particular, a valid generalization G is represented by a set of *generalization nodes* S_G in the domain hierarchy tree that satisfy the following condition: *The path from every leaf to the root along the tree encounters one (to guarantee generalizability) and only one (to guarantee deterministic generalization) generalization node in* S_G. This definition includes the case of a leaf node itself being a generalization node. We have seen domain of a categorical attribute being organized into a domain hierarchy tree (e.g., Figure 6.1); we next describe the generalization of a *numeric column*. It is accomplished by first dividing the domain space of the column into a series of disjoint intervals, and then pairwise combining them into a binary tree. With the tree, generalization proceeds in the same way as for a categorical attribute. As an example, Figure 6.3 depicts the construction of a *binary* domain hierarchy tree for the column Age with domain [0, 150). In order to avoid over-binning the data, intervals should be of *moderate* size (smaller) and they need not to be of equal size.

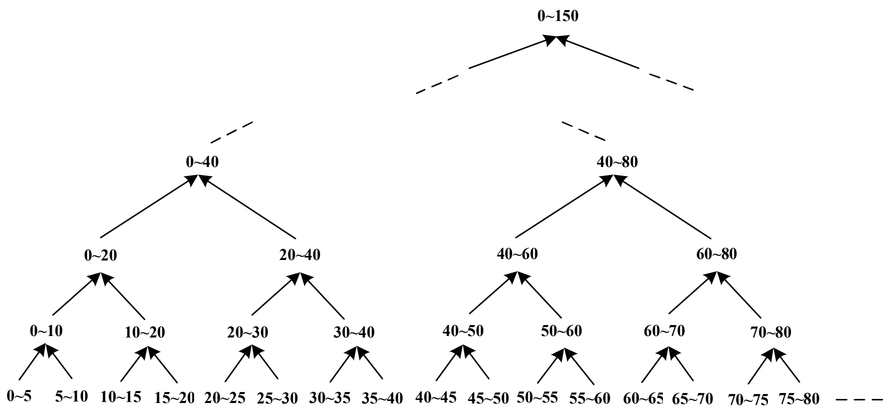


Figure 6.3: Constructing Binary DHT for A Numeric Attribute.

Clearly, binning makes data less specific and more general, thereby resulting in some information loss. It would make no sense to meet k-anonymity specification if that renders the data useless, thus data quality must be preserved. We suggest constraining the binning process to abide by *usage metrics* specifying a set of maximal allowable information loss. More information loss than as specified would substantially degrade the data quality with respect to the intended data usage.

6.4.1 Usage Metrics

Consider first a categorical column c that associates with a domain a hierarchy tree T, e.g., Figure 6.1. If *Pharmacist* is generalized to *Paramedic*, under our definition of generalization, child nodes of *Paramedic* would become indiscriminatable. This in turn implies that all entries in c containing *Pharmacist/Nurse/Consultant* would become indiscriminatable. This concept of indiscrimination leads to our approach for quantifying information loss $InfLoss_c$ for the column c as follows. Suppose a generalization results in a set of generalization nodes $\{p_1, p_2, ..., p_M\}$; let S_i be a set containing the leaf nodes of the subtree that is rooted at p_i, and the number of entries in c containing values in

S_i be n_i, $i = 1..M$. Information loss $InfLoss_c$ is defined as

$$InfLoss_c = \frac{\sum_{i=1}^{M}(n_i\frac{|S_i|-1}{|S|})}{\sum_{i=1}^{M} n_i} \tag{6.1}$$

where $S = S_1 \bigcup S_2 \bigcup ... \bigcup S_M$ is the set of leaf nodes of the tree T. We allow some leaf nodes to remain ungeneralized given that k-anonymity specification is already met, in which case $|S_i| = 1$.

We next consider a numeric attribute c, e.g., Age. Suppose the domain of c, whose lower and upper bounds are L and U, respectively, is generalized into M intervals. The lower and the upper bounds for these intervals are L_i and U_i, respectively, $i = 1..M$. Let n_i be the number of entries in the column c whose values fall between L_i and U_i, $InfLoss_c$ is then defined as

$$InfLoss_c = \frac{\sum_{i=1}^{M}(n_i\frac{U_i-L_i}{U-L})}{\sum_{i=1}^{M} n_i} \tag{6.2}$$

Once all $InfLoss_i$, $i = 1..CN$ (CN is the total number of the columns to be generalized) are determined, a normalized loss $InfLoss$ is computed by averaging over all generalized columns in the table:

$$InfLoss = \frac{\sum_{i=1}^{CN} InfLoss_i}{CN} \tag{6.3}$$

Likewise, other forms of information loss, e.g., total information loss can be defined. Finally, based on the definition of information loss, usage metrics for controlling information loss are defined in general as following:

$$InfLoss_i \leq bd_i \;\; \forall i = 1, ..., CN \tag{6.4}$$

$$InfLoss \leq bd_{avg}$$

where $\mathcal{B} = \{bd_1, ..., bd_{CN}\} \subset \mathcal{R}$ and $bd_{avg} \in \mathcal{R}$ define the bounds for maximal allowable information loss.

In practice, the enforcement of the above metrics in a normal way might not be ideal as it involves calculating information loss and in turn checking against the bounds after every step of binning. Fortunately, we can implement an off-line enforcement, yielding a set of *maximal generalization nodes* in each domain hierarchy tree. Maximal generalization nodes are defined as 1) they constitute a valid generalization; 2) each of them being the highest node in the domain hierarchy tree to which the corresponding leaf nodes can be generalized under the usage metrics. Usage metrics in the form of maximal generalization nodes are obviously much easier to enforce, only requiring that none of the leaf nodes be generalized beyond its corresponding maximal generalization node. It is preferable that the maximal generalization nodes are directly given as the usage metrics, rather than being transformed from the form of Equation 6.4.

We note that a generalization comprising the maximal generalization nodes trivially satisfies k-anonymity specification given that the data are *binnable*. The point is to meet k-anonymity while minimizing information loss. It is thus clear that binning would yield a set of generalization nodes that are lower than or at most equal to the maximal generalization nodes. This reasonably reflects the underlying principle that binning is not allowed to damage data usefulness. Let us consider the earlier example of generalizing a numeric attribute, where we suppose the set of intervals in satisfying k-anonymity is depicted by the leaf nodes of the tree in Figure 6.4: enforcement of the usage metrics might most likely allow for further generalizations, yielding the set of maximal generalization nodes denoted as elliptic nodes.

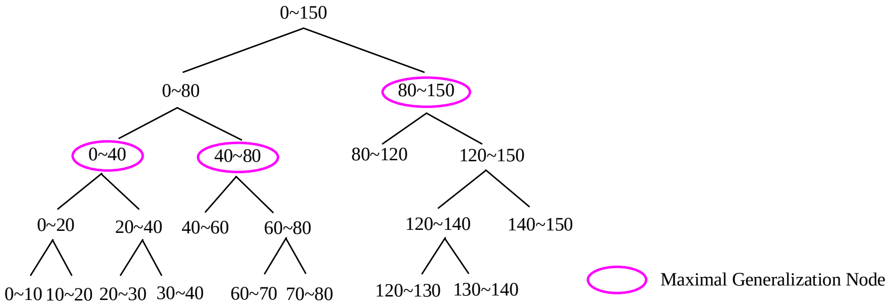


Figure 6.4: A DHT by Enforcing Usage Metrics.

6.4.2 Binning

We decompose binning into two steps, i.e., mono-attribute binning and multi-attribute binning. The mono-attribute binning step bins attributes individually so that each transformed attribute satisfies k-anonymity. The multi-attribute binning step is required because, while each attribute satisfies k-anonymity, combinations of them may not. Consider an example of a transformed table, where 36 people have an age between $25 \sim 50$ and 8 people are doctors, each satisfying k-anonymity specification with $k = 6$. However, there might be only 4 people who are aged between $25 \sim 50$ who are also doctors.

For ease of referencing, we list in Table 6.1 the variables and functions that will be used in this and the next section.

Notation	Meaning
tr	the domain hierarchy tree for an attribute
tbl	the table to be protected
$mingends$	the set of minimal generalization nodes
$maxgends$	the set of maximal generalization nodes
$ultigends$	the set of ultimate generalization nodes
k	the system parameter for k-anonymity
k_1, k_2, η	elements of the secret watermarking key
wm, wmd	actual and replicated mark, respectively
Parent(nd, tr)	returns the parent node of nd in tr
Children(nd, tr)	returns the set of child nodes of nd in tr
Siblings(nd, tr)	returns nd together with its sibling nodes in tr
Leaves(tr)	returns the set of leaf nodes of tr
SubTree(nd, tr)	returns the subtree of tr rooted at nd
Duplicate(wm)	duplicates wm to produce wmd
Val2Nd(v, nds[])	returns the node in nds[] that represents v
Nd2Val(nd)	returns the value represented by nd
SetμBit(v, b)	sets the least significant bit of v to be the bit b
Index(nd, S)	returns the index of nd in the set S
MajorVot(wmd)	majority voting over wmd

Table 6.1: Variables and Functions

6.4.2.1 Mono-attribute Binning

For an individual attribute, our binning starts from the maximal generalization nodes downwards along the domain hierarchy tree, until reaching a set of lowest nodes that constitute a valid generalization catering to k-anonymity specification. We term such nodes *minimal generalization nodes*. Our way of downward binning is an advantage offered by the off-line enforcement of usage metrics. The mono-attribute binning is basically an *exhaustive* trial procedure in a search for the minimal generalization nodes. For this reason, compared to previous work that bins upward along the tree (e.g., [118]), downward binning turns out to be more efficient. The intuition is that the higher level on the tree, the less nodes are to be tried. Note that the observance of usage metrics is directly accomplished by starting binning from the maximal generalization nodes.

Figure 6.5 outlines the algorithm for generating the set of minimal generalization nodes.

```
**GenMinNd**($tr$, $maxgends$, $tbl$, $k$)
    $mingends \leftarrow$ NULL
    **foreach** node $nd \in maxgends$
        $subtr \leftarrow$ SubTree($nd$, $tr$)
        $mingends \leftarrow mingends \bigcup$ SubGMN($subtr$, $tbl$, $k$)

**SubGMN**(tree $str$, $tbl$, $k$)
    **if** NumTuple($str$, $tbl$) $< k$
        **return** NULL
    **forany** node $nd \in$ Children($str$.root, $tr$)
        **if** NumTuple(SubTree($nd$, $str$), $tbl$) $< k$
            **return** $\{str$.root$\}$
    $tmpset \leftarrow$ NULL
    **foreach** $nd \in$Children($str$.root, $str$)
        $subtr \leftarrow$ SubTree($nd$, $str$)
        $tmpset \leftarrow tmpset \bigcup$ SubGMN($subtr$, $tbl$, $k$)
    **return** $tmpset$

**NumTuple**(tree $str$, $tbl$)
    int $num = 0$
    **foreach** tuple $t_i \in tbl$
        **if** $t_i$.val $\in$ Leaves($str$)
            $num \leftarrow num + 1$
    **return** $num$
```

Figure 6.5: Mono-attribute Binning Algorithm

We employ a simple rationale in generating a minimal generalization node: a node is *minimal* if itself meets k-anonymity, but not all of its child nodes do. This might lead to an over-generalization of the data. A more aggressive strategy could be enlisted, e.g., a node is not minimal if *any* of its child nodes satisfies k-anonymity.

6.4.2.2 Multi-attribute Binning

Multi-attribute binning involves further binning attributes, each of which already satisfies k-anonymity. However, for an individual attribute, the set of allowable generalizations for the purpose of multi-attribute binning is already defined by the nodes between the minimal generalization nodes and the maximal generalization nodes. Consider Figure 6.6: the set of allowable generalizations constrained by the minimal generalization nodes and the maximal generalization nodes are enumerated as {30, 31, 45, 46, 33, 22}, {30, 31, 32, 33, 22}, {30, 31, 21, 22}, {20, 45, 46, 33, 22}, {20, 32, 33, 22} and {20, 21, 22}. As a result, the set of allowable generalizations for the entire table is the enumeration of different combinations of allowable generalizations for all attributes. Let the number of quasi-identifying columns be CN, and n_i be the number of allowable generalizations for column i, then the total number of allowable generalizations for the table is $\prod_{i=1}^{CN} n_i$.

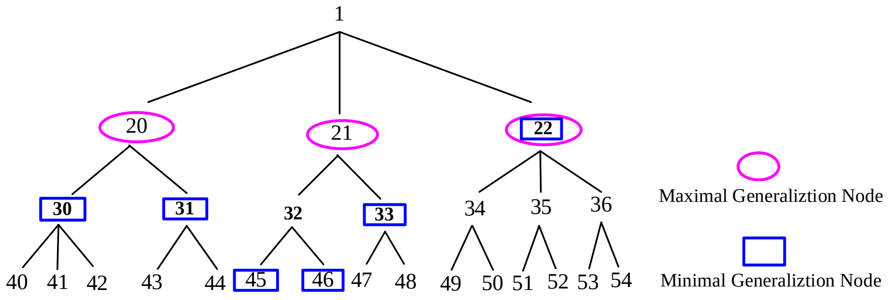


Figure 6.6: A DHT for Illustrating Multi-attribute Binning.

Among these allowable generalizations, some do not satisfy k-anonymity, and are thereby invalid; the remaining are valid for k-anonymity. Nevertheless, not all these valid generalizations are equally satisfactory. The point here is to choose among them an *ultimate generalization* that results in the minimal information loss. Nodes in this ultimate generalization are called *ultimate generalization nodes*. Clearly, the calculation of information loss can be done by using Equation (6.1), (6.2) and (6.3), although this may not be ideal as it may incur unacceptable computation penalty. Instead, we prefer simplifying this calculation by solely considering "specificity loss" regarding the domain hierarchy trees. Let the total number of leaf nodes of a tree be N and the number of generalization nodes of an allowable generalization be N_g, we define specificity loss due to generalization to be $(N - N_g)/N$. This approach of estimating specificity loss results in a more efficient implementation, but it may reduce accuracy.

Figure 6.7 outlines the above approach for determining the ultimate generalization nodes. The function EnumGen(.) enumerates all distinct combinations of allowable generalizations among attributes, and the function Selection(.) determines the generalization that incurs least specificity loss.

```
**GenUltiNd**($mingends$[1..CN], $maxgends$[1..CN], $tr$[1..CN])
    **for** $i$ = 1..CN
        $allowblgens[i] \leftarrow \{gen_j \mid gen_j$ is a generalization
            constrained by $mingends[i]$, $maxgends[i]$ in $tr[i]\}$
    $allgens \leftarrow$ EnumGen($allowblgens[i]$, $i$ = 1..CN)
    $validgens \leftarrow \{gen_j \mid gen_j \in allgens \bigwedge gen_j$ satisfies
        $k$-anonymity$\}$
    $ultigen \leftarrow$ Selection($validgens$)
```

Figure 6.7: Multi-attribute Binning Algorithm

6.4.2.3 Binning Algorithm

A relevant observation to make is that the identifying columns are most likely to be the key attributes (e.g., primary key) of the table, containing the most important part of information. Hence it is frequently useful to maintain the identifying columns traceable to the data holder in health care domain. For instance, as reported in [67], in some cases patients may benefit from being traced in research such as the assessment of treatment safety. Moreover, many real-world clinical projects such as those in [111] and in [68] support traceability of the medical data. Based on this observation, our binning algorithm adopts an one-to-one replacement for data in the identifying columns. In particular, we replace each data by its encrypted value that is generated by an encryption function $\mathcal{E}()$ e.g., AES. We point out that keeping the identifying columns unsuppressed and unmanipulated further is also important for watermarking. Figure 6.8 outlines our complete binning algorithm, comprising the encryption of the identifying columns and the binning of the quasi-identifying columns. Given the ultimate generalization *ultigen* yielded by multi-attribute binning, the function Bin(.) works by simply replacing each value in the quasi-identifying columns by the value represented by its corresponding node in *ultigen*.

```
Binning($tbl$, $ultigen$)
    foreach tuple $t_i \in tbl$
        $t_i$.ident.val $\leftarrow \mathcal{E}(t_i$.ident.val$)$
        $t_i$.quasi-ident.val $\leftarrow$ Bin($t_i$.quasi-ident.val, $ultigen$)
```

Figure 6.8: Binning Algorithm

6.5 Watermarking Algorithm

By its very nature, watermarking modifies the data to be watermarked, thereby further degrading data quality. Watermarking works under a general assumption that the underlying data can tolerate a certain degree of quality degradation. The tolerance closely relates to the bandwidth for insertion, implying that watermarking would fail unless the data can be modified. The discovery of the available bandwidth appears to be challenging in the case of watermarking relational data [156, 171]. We next explain how to find the desired bandwidth channel for insertion in the binned data.

6.5.1 Bandwidth Channel

In our context, columns of a table after binning become essentially categorical, and data modification by watermarking is equivalent to the permutation of data. We advocate that a binned table can actually accommodate some degree of data permutation, thereby providing the desired bandwidth channel for watermarking.

From earlier discussions, we know that generalization of a node in the hierarchy tree to its parent node renders indiscrimination among this node and its sibling nodes. In essence, a random permutation of values represented by these nodes equals the effect of the generalization. As long as such a generalization is allowed, watermarking relying on the data permutation would definitely work. Recall that the set of maximal generalization nodes defined by usage metrics are normally atop the set of ultimate generalization nodes resulting from binning. Hence, generalizations between the two levels still respect usage metrics, which in turn guarantee the viability of watermarking. It is important to notice a special case where a ultimate generalization node itself is also a maximal generalization node. Permutation of such nodes might result in information loss above the threshold set by usage metrics. However, watermarking affects only a small fraction of the data set, and hence such excessive loss is expected to be minor. As a matter of fact, this is the price that any watermarking must pay. More importantly, we can readily tackle this scenario by slightly modifying the way a maximal generalization node is defined. Specifically, in determining the set of maximal generalization nodes, the bounds in Equation 6.4 are given slightly lower than actually required for sustaining data usage, so that a small fraction of the table is allowed to be generalized to the values represented by the maximal generalization nodes. Note however that such transformation on a large scale would definitely destroy the data.

6.5.2 Watermarking at A Single Level

A direct way to take advantage of the above bandwidth channel is to consider permutation at the level of each ultimate generalization node (together with its sibling nodes). The exact primitive enabling bit insertion works as follows. Suppose an ultimate generalization node p needs to be permutated, and p and its sibling nodes compose a sorted set S. To insert a bit b, our basic idea for determining a target node q in S such that $p \rightarrow q$ encodes the bit b is: the index of q in S is even, if $b = 0$; the index of q in S is odd, if $b = 1$. However, this does not suffice since some elements in S may not be ultimate generalization nodes, so if the target node q is not an ultimate generalization node, validity of the generalization (see Section 6.4) is violated. To solve this issue, we shall continue the permutation process downward among the child nodes of q, and possibly even lower, until an ultimate generalization node is reached. Our definition of *generalization* guarantees the reachability. This idea of achieving embedding by data permutation is similar to [171], but we do within finer domains (sub-domain of the column), and more importantly we have solid justifications for permutation. Unfortunately, watermarking at this single level is susceptible to a kind of *generalization attack* that can completely destroy the inserted bits without knowing the watermarking key.

6.5.2.1 Generalization Attack

The generalization attack is specific to the binned data. It works as follows: the attacker starts a further generalization on the watermarked table, generalizing each value to the value represented by a higher generalization node in the domain hierarchy tree. Because of the gap between the maximal generalization nodes and the ultimate generalization nodes, the table would sustain data usage. The generalization attack appears fatal as it does not require the secret watermarking key at all. A careful analysis indicates that it is the way we consider watermarking only at the level of ultimate generalization nodes that makes possible the attack. To thwart this attack, we must additionally watermark all intermediate levels between the maximal generalization nodes and the ultimate generalization nodes. This constitutes the basic idea of our hierarchical watermarking scheme.

6.5.3 A Hierarchical Watermarking Scheme

In the hierarchical watermarking, we consider watermarking at every level, from the maximal generalization nodes to the ultimate generalization nodes. Specifically, for an ultimate generalization node p to be permutated, watermarking starts by first determining the maximal generalization node q that corresponds to p, followed by executing permutations downward along the domain hierarchy tree from the level of the child nodes of q, until the target node is an ultimate generalization node. The exact primitive enabling permutation at each level is the same as above. Consider Figure 6.6 for example (for illustration's sake, we need to intentionally take the minimal generalization nodes therein as the ultimate generalization nodes), where node 46 is going to be permutated. First, the corresponding maximal generalization node 21 is determined. Next, permutation proceeds within nodes 32 and 33. If the target node is node 33, then permutation stops; otherwise, the permutation continues within nodes 45 and 46, and eventually stops.

To avoid a large scale alteration, watermarking is ideally restricted to a (small) portion of the whole data set. We leverage on the (encrypted) identifying columns of the binned table to select some tuples for embedding, recalling that the encrypted identifying columns are assumed to keep intact[1]. Based on a secret key k_1 together with a secret tunable parameter η, tuples t_i in the table *tbl* satisfying the following equation are chosen for insertion:

$$\mathcal{H}(t_i.\text{ident},\ k_1) \bmod \eta = 0 \quad \forall t_i \in \mathit{tbl} \tag{6.5}$$

where $\mathcal{H}()$ is a cryptographic hash function e.g., MD5 or SHA1, and *tbl*.ident denotes the encrypted identifying columns of *tbl*. Note that the way of secretely selecting tuples directly pertains to the resilience of watermarking.

Typically, the available bandwidth is greater than the bit length $|wm|$ of the mark wm. This affords a multiple embedding of wm for robustness reasons. That is, we repeatedly embed wm many times until the available bandwidth is exhausted. In mark detection phase, the final mark is determined by *majority voting* over all the recovered copies. A straightforward way to achieve multiple embedding is to duplicate wm for l times into wmd, as long as we attempt an l-embedding, and then to insert wmd in place

[1] In case the identifying columns cannot be relied on, we can establish virtual key attributes as in [123] by turning to other columns

of *wm*.

Take *tbl*.c, a quasi-identifying column of *tbl* for example, our hierarchical watermarking algorithm by integrating the above ideas, is outlined in Figure 6.9. The function MaxGNd(*nd*, *tr*, *maxgends*) returns the maximal generalization node that associates with *nd*.

```
**Embedding**($tbl$, $tr$, $maxgends$, $ultigends$, $k_1$, $k_2$, $\eta$, $wm$)
    bits $wmd \leftarrow$ Duplicate($wm$)
    **foreach** tuple $t_i \in tbl$
        **if** $\mathcal{H}(t_i$.ident, $k_1)$ mod $\eta = 0$
            node $targnd \leftarrow$ Val2Nd($t_i$.c, $ultigends$)
            $targnd \leftarrow$ MaxGNd($targnd$, $tr$, $maxgends$)
            **do**
                $targnd \leftarrow$ Permutate($targnd$, $tr$, $t_i$, $k_1$, $k_2$, $wmd$)
            **while** $targnd \notin ultigends$
            $t_i$.c $\leftarrow$ Nd2Val($targnd$)
```

```
**Permutate**(node $nd$, $tr$, tuple $t_i$, $k_2$, bits $wmd$)
    sortedset $S \leftarrow \{s_i \mid s_i \in$ Children($nd$, $tr$)$\}$
    int $indx \leftarrow \mathcal{H}(t_i$.ident, $k_2)$ mod $|S|$
    $indx \leftarrow$ Set$\mu$Bit(indx, $wmd[\mathcal{H}(t_i$.ident, $k_2)$ mod $|wmd|]$)
    **return** $s_{indx}$
```

```
**Detection**($tbl$, $tr$, $maxgends$, $ultigends$, $k_1$, $k_2$, $\eta$, $wm$)
    bits $wmd \leftarrow$ NULL      /* set $wmd$ to be empty */
    **foreach** tuple $t_i \in tbl$
      **if** $\mathcal{H}(t_i$.ident, $k_1)$ mod $\eta = 0$
          node $tmpnd \leftarrow$ Val2Nd($t_i$.c, $ultigends$)
          bit[] b = NULL, int $i = 0$      /* reset */
          **do**
              sortedset $S \leftarrow \{s_i \mid s_i \in$ Siblings($tmpnd$, $tr$)$\}$
              int $indx \leftarrow$ Index($tmpnd$, $S$)
              $b[i] \leftarrow indx$&1
              $i \leftarrow i + 1$
              $tmpdnd \leftarrow$ Parent($tmpnd$, $tr$)
          **while** $tmpnd \notin maxgends$
          $wmd[\mathcal{H}(t_i$.ident, $k_2)$ mod $|wmd|] \leftarrow$ MajorVot($b$)
     $wm \leftarrow$ MajorVot($wmd$)
```

Figure 6.9: Hierarchical Watermarking Algorithm

In the algorithm, we exploit distinct keys k_1 and k_2 for different calculations, which is vital in ensuring that there is no mutual correlation between these calculations. Notice that the hierarchical scheme enables to insert several copies of a bit at every single embedding position, and the actual number is equal to the number of levels from the corresponding maximal generalization node to the ultimate generalization node. Thus, when recovering a bit from a single embedding position, the bit is determined by majority voting. Interestingly, in the voting process, we can assign a different weight to each copy from a distinct level, depending on its credit in determining the bit. This is of special use when enforcing the policy that the copy from a higher level is more reliable than that from a lower level.

6.5.4 Resolving Rightful Ownership Problem

Robustness to attacks attempting to erase the embedded mark is among the fundamental requirements of a sound watermarking. However, this does not necessarily imply its sufficiency in establishing ownership, because of the attacking scenarios in Figure 6.10 (D_x, W_x and K_x are respectively the original data, the mark and the secret watermarking key of the entity x, D_w and $\overline{D}_w$ denote the watermarked data).

Attack 1: the attacker inserts his bogus mark W_a into D_w, which is the owner's valid watermarked data, to create a bogus $\overline{D}_w$. Now that both W_o and W_a are contained in $\overline{D}_w$, the attacker and the owner can both claim the ownership over $\overline{D}_w$. This attack can be resolved by requiring the attacker and the owner each to present his original data. As the attacker's "original" data D_w contains W_o of the owner, false ownership claim by the attacker is clear.

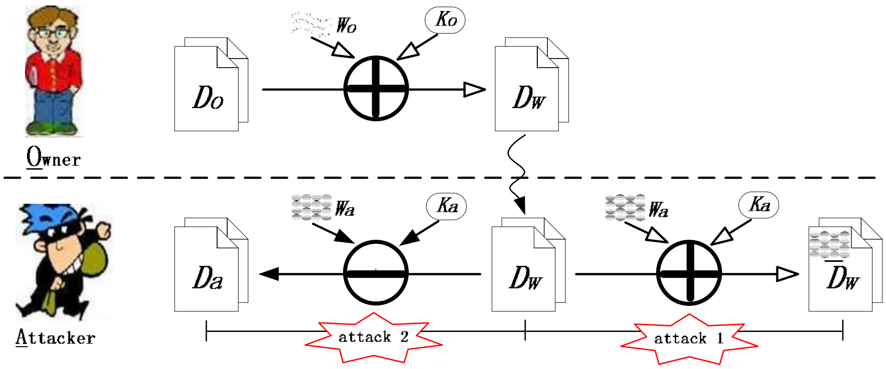


Figure 6.10: Rightful Ownership Attacks.

Attack 2: In this case however, the attacker "extracts" W_a from D_w to obtain his bogus original data D_a, so that $D_a \oplus_{k_a} W_a = D_w$, where $\oplus_{k_a}$ denotes the embedding function under key k_a. This attack is more subtle to handle, since it does not always hold that D_a contains W_o and D_o does not contain W_a. So far, the only practical solution in multimedia watermarking is to restrict W_o to be $\mathcal{F}(D_o)$, where $\mathcal{F}(.)$ is an one-way function, so that given D_w, it is impossible to acquire D_a by the attacker satisfying $D_a \oplus_{k_a} W_a = D_a \oplus_{k_a} \mathcal{F}(D_a) = D_w$.

These attacks are in fact the rightful ownership problem originally raised in [169] in multimedia context. It will be of particular interest to see how the rightful ownership problem is handled in our case. We notice that virtually none of the existing proposals for watermarking databases has provided a satisfactory solution to this problem, as either they considered merely one case of it (e.g., [7, 123]) or they did not address it at all (e.g., [156,171]). Results from the multimedia sector show that without invoking a third party for certifying the watermarked data D_w, the rightful ownership problem is solvable only when the original data are available in court. We believe this directly applies to the context of databases. Considering the large number of data a table contains, we actually suspect the practicality of presenting to the judge the entire original table as court proof in other proposals. Surprisingly, the nature of the binned data enables us to elegantly resolve this problem in our context. Recall that the identifying columns of a binned table to be watermarked are in encrypted format, which means the attacker has no way to know the clear-texts. So the mark in our scheme is specified by applying the one-way function $\mathcal{F}(.)$ to a certain statistical value v (e.g., mean) of these clear-texts of the identifying columns (the attacker cannot get $\mathcal{F}(v)$ or forge other valid marks). In resolving ownership dispute, the owner presents v; decrypts the identifying columns and does the same statistical computation over the decrypted data to get v'; compares the two as valid if $|v - v'| < \tau$, where τ is a predefined threshold; extracts the mark from the table in dispute and compares it with $\mathcal{F}(v)$ as usual in a normal watermarking scheme. Note that most probably, the watermarked table in dispute had been attacked, e.g., some tuples were deleted or some spurious tuples were added, and this explains why we acquire the mark from a statistical value instead of the actual clear-texts.

The proposed solution is specific to our integration of binning and watermarking, since a normal database does not have encrypted attributes as in our case (in case the identifying columns are not encrypted, attackers can easily derive other marks). In nature, we do not violate "original data as court proof", whereas the integrated property of our framework provides an effective means to get over direct reliance on the entire original table, but on a statistic value of the clear-texts of the encrypted columns.

6.6 Analysis

We next explore the seamlessness of our framework from a theoretical perspective. In other words, we are concerned with the effect watermarking has on the result of binning. The main issue is related to the fact that watermarking in our context involves permutation such that some tuples in a bin may be permutated to other bins, and thus some bins may have, after watermarking, a size less than k. This means that watermarking may compromise the satisfaction to k-anonymity of binning. Without loss of generality, we restrict our discussions to a particular quasi-identifying column c, which corresponds to a domain hierarchy tree having m maximal generalization nodes N_i $(i = 1..m)$, and n_i ultimate generalization nodes associated with each node N_i. We further make the following assumptions: (i) bins that correspond to the ultimate generalization nodes are of equal size; (ii) when a bit-embedding proceeds downward from N_i, all the n_i ultimate generalization nodes associated with N_i have equal probability of becoming the target node when permutations halt. The actual effect of watermarking on binning can be reduced to the way any particular bin (BIN) that corresponds to a ultimate generalization node UGN is affected by any bit-embedding (E).

Lemma 6.1 *Let the maximal generalization node corresponding to UGN be* N_k*, and the probability of E reducing the bin size of BIN by* 1 *be* Pr^-*, then* $Pr^- = \frac{n_k - 1}{n_k \sum_{i=1}^{k} n_i}$.

Proof: Intuitively, for E to reduce the bin size of BIN by 1, it must hold that as per our hierarchical watermarking algorithm, 1) the bit chosen by E for insertion comes from BIN; 2) afterwards, E executes downward permutations (starting from N_k) among the n_k ultimate generalization nodes that correspond to N_k, and the target node of such permutations is not UGN. From assumption (i), probability that the tuple chosen by E comes from BIN is $\frac{1}{\sum_{i=1}^{m} n_i}$, and from assumption (ii), probability of the target node not being UGN is $\frac{n_k - 1}{n_k}$. Hence, altogether $Pr^- = \frac{1}{\sum_{i=1}^{m} n_i} \times \frac{n_k - 1}{n_k} = \frac{n_k - 1}{n_k \sum_{i=1}^{k} n_i}$. □

Lemma 6.1 states the probability of any particular bit-embedding E permutating a tuple out of a particular bin BIN. We next check the probability of E permutating a tuple from another bin to BIN.

Lemma 6.2 *Let the maximal generalization node corresponding to UGN be* N_k*, and the probability of E increasing the bin size of BIN by* 1 *be* Pr^+*, then* $Pr^+ = \frac{n_k - 1}{n_k \sum_{i=1}^{k} n_i}$.

Proof: For E to increase the bin size of BIN by 1, it must hold that 1) E selects the tuple for insertion from any, but UGN, of the n_k ultimate generalization nodes that are associated with N_k ; 2) the target node of the downward permutations is UGN. From assumption (i), probability of the former is $\frac{n_k - 1}{\sum_{i=1}^{m} n_i}$, and from assumption (ii), probability of the latter is $\frac{1}{n_k}$. Hence, $Pr^+ = \frac{n_k - 1}{\sum_{i=1}^{m} n_i} \times \frac{1}{n_k} = \frac{n_k - 1}{n_k \sum_{i=1}^{k} n_i}$. □

Lemma 6.1 and Lemma 6.2 suggest that on average, the watermarking process would neither decrease nor increase the bin size of any bin since $Pr^- = Pr^+$. We therefore conclude that watermarking does not interfere with binning in the satisfaction of k-anonymity specification under the two ideal assumptions.

It is of importance to examine the assumptions from a practical perspective. Making valid the first assumption is not that hard: we can incorporate "restrained swapping" (e.g., swapping tuples among bins that correspond to sibling nodes) into binning. In contrast, the second assumption is more tricky, because its validity totally rests with the locality of ultimate generalization nodes on the domain hierarchy tree. Even so, we believe that by relaxing the two assumptions, watermarking still cannot seriously interfere with binning because: 1) only a small percentage of the whole data gets watermarked; 2) and the use of hash function in the "suitability" selection step (Equation 6.5) renders a uniform culling, which means no particular bin will be drastically affected. To attest this, we have done experiments and obtained consistent results (see next section). After all, we have a simple yet practical method to tackle the interference by applying $k + \epsilon$ (ϵ is a small number) to binning in meeting k-anonymity specification.

6.7 Experimental Studies

We implemented and conducted extensive experiments on the above algorithms. The real world data set we experimented on include one (randomized) identifying column and five quasi-identifying columns, whose schema is $\mathcal{R}$(*ssn*, *age*, *zip_code*, *doctor*, *symptom*, *prescription*). By a preprocessing step, we created a DHT for each quasi-identifying column: the DHT for *symptom* is based on the International Classification of Diseases (ICD-9), and other attributes are on self-defined ontology, e.g., that for *age* is similar to Figure **??** but of narrower intervals. The whole data set contains around 20000 tuples. Experiments were done on a PC with 2G CPU and 512M RAM, and source codes were written in Microsoft C++. A main *simplification* we made is that a set of maximal generalization nodes is directly given to each column as usage metrics.

6.7.1 Robustness of Binning

First, our experiments focus on testing the binning algorithm in satisfying k-anonymity. By providing to the algorithm different values of k, we recorded the corresponding loss of information. Figure 6.11 shows the relationship of k versus information loss.

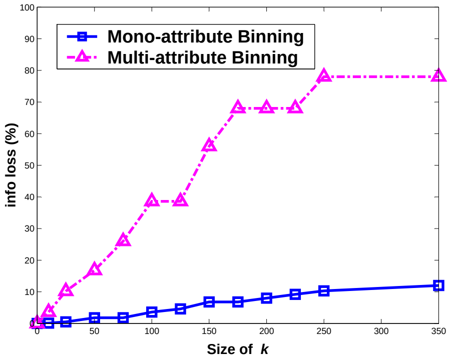


Figure 6.11: k vs. Information Loss.

From the figure, multi-attribute binning causes much more information loss than mono-attribute binning, and once k increases to a certain extent, information loss reaches a saturation point and becomes rather stable. This is consistent with the rationale in determining a valid minimal generalization node (Section 4.2), and this could be further optimized if the more aggressive strategy as introduced there is employed. Further, we should also note that information loss is closely related to the data size, the number of quasi-identifying columns and k.

6.7.2 Robustness of Watermarking

In this set of experiments, we test the robustness of the hierarchical watermarking scheme to the attacks that endeavor to destroy the embedded mark, while in the absence of the secret watermarking key. The following experiments were conducted by implementing a multiple embedding of a 20-bit mark.

- *Subset Alteration*

In these attacks, the attacker chooses at random a subset of the data and then modifies them arbitrarily without affecting the rest of the data. We vary the size of the randomly altered data, and calculate the corresponding mark loss. Figure 6.12 (a) outlines the results. Clearly, the results show that our watermarking scheme performs well against this attack. Even in the case of more than 70% of data loss, our scheme loses only approximately 30% of mark bits. Another fact shown in the figure is that smaller η (more bandwidth) offers more resilience, whereas more alteration to the data would be incurred. This is a trade-off that must be carefully considered in practice.

- *Subset Addition*

In these attacks, new tuples are frequently added to the watermarked set by the malicious attacker. Although this attack does not involve erasing existing bits, it nevertheless misleads the selection criteria (Equation (6.5)) to falsely take some of the newly-added tuples as watermarked, thereby introducing errors in majority voting the final mark. Keep in mind that if the size of the new data exceeds the original data size, priority of the former would dominate the latter. Figure 6.12 (b) highlights the scheme's robustness to the Subset Addition attacks. The results reflect the fact that the newly-added bogus bits do not take precedence over the existing bits in the majority-voting process.

- *Subset Deletion*

The attacker randomly deletes a percentage of the tuples in an attempt to remove the mark. To test the effect of dropping tuples to the loss of mark bits, we continually delete some tuples each time by the following SQL clause:

DELETE FROM $\mathcal{R}$ WHERE SSN > lval_i AND SSN < uval_i

where lval_i and uval_i define bounds of the i^{th} deletion, within which the tuples are to be deleted. Figure 6.12 (c) plots the series of mark loss due to the deletions. From the figure, it indicates that the hierarchical scheme is resilient to the Subset Deletion attacks, and mark loss increases almost linearly with the amount of data deleted.

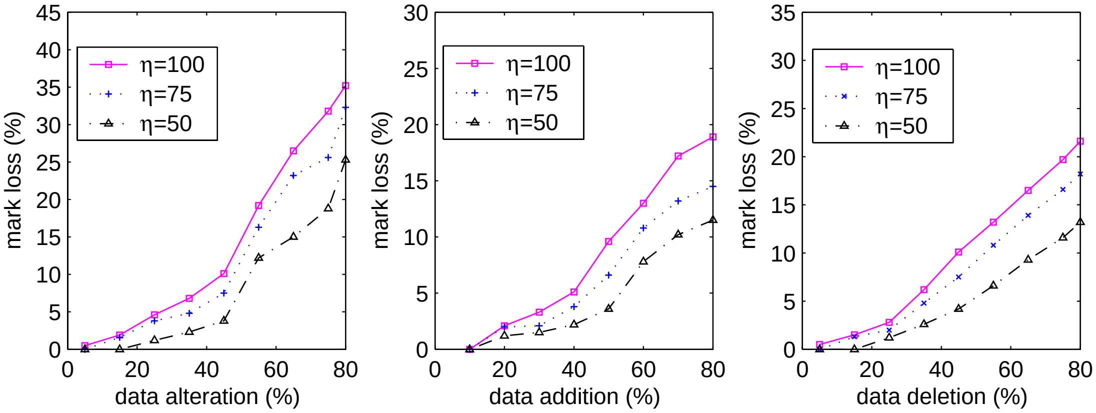


Figure 6.12: Robustness of Hierarchical Watermarking.

We also tested the information loss due to watermarking, and Figure 6.13 presents the results. Clearly, information loss caused by watermarking is minor.

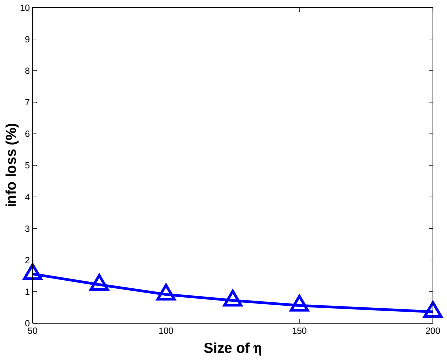
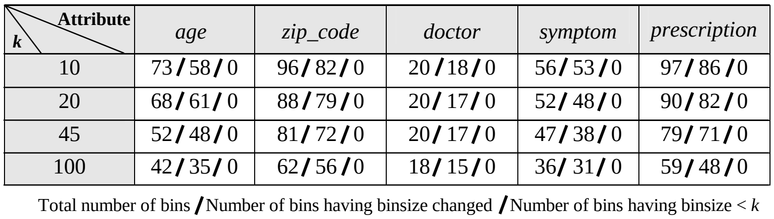


Figure 6.13: Information Loss of Watermarking.

6.7.3 Seamlessness of Framework

Finally, we shall examine how watermarking interferes with binning, complementing the theoretic analysis in the preceding section. The results are presented in Figure 6.14, where the data in each column respectively represents the total number of bins, number of bins having bin size changed and number of bins having bin size less than k. It can been seen that a majority of the bins are affected by watermarking, whereas the interference is minor in terms of satisfying k-anonymity: none of the bins cannot meet k-anonymity after watermarking. This is consistent with our analysis that watermarking does not dramatically affect binning in its compliance with k-anonymity specification.



Attribute *k*	*age*	*zip_code*	*doctor*	*symptom*	*prescription*
10	73 / 58 / 0	96 / 82 / 0	20 / 18 / 0	56 / 53 / 0	97 / 86 / 0
20	68 / 61 / 0	88 / 79 / 0	20 / 17 / 0	52 / 48 / 0	90 / 82 / 0
45	52 / 48 / 0	81 / 72 / 0	20 / 17 / 0	47 / 38 / 0	79 / 71 / 0
100	42 / 35 / 0	62 / 56 / 0	18 / 15 / 0	36 / 31 / 0	59 / 48 / 0

Total number of bins / Number of bins having binsize changed / Number of bins having binsize < *k*

Figure 6.14: Effect of Watermarking on Binning.

6.8 Concluding Remarks

Two important issues inherent to the outsourcing of health data are the protection of individual privacy and copyright protection over the data. To meet these dual needs, we integrated techniques of binning and digital watermarking into a unified framework, so as to provide comprehensive protection for outsourced data. Under our framework, health data are in turn binned to meet k-anonymity specification, and watermarked to provide copyright protection. We have discussed at length the development of the binning algorithm and the watermarking algorithm that provide the two core functions in our framework, and developed an elegant solution to the rightful ownership problem regarding watermarking, which may be difficult to solve in the context of other approaches. From both theoretical and practical perspectives, we proved that watermarking would not substantially interfere with binning in the satisfaction to k-anonymity. Experimental results showed the robustness of the proposed framework.

CHAPTER 7

Conclusions and Future Work

We systematically studied data security and especially individual privacy in health care systems, focusing on the following closely related issues in particular.

We first discussed building a unified trust infrastructure for individual health care organizations. To that end, each organization establishes a dedicated security manager for handling security related matters such as certification, issuance of secret keys. The organizational trust infrastructure is thus built around the security manager, by incorporating various user authentication techniques and modes such as short password, identity certificate, attribute-based certificate, anonymous credential, and group signature. Apart from password, all other authentication techniques can directly enlist the security manager as the CA or the TTP, we thus focused on unifying password authentication within the trust infrastructure. Our solution was a novel two-server password authentication system that exploits the security manager operating a back-end authentication server for assisting the service server in user authentication. Our proposed two-server password authentication system can also circumvent weaknesses inherent in the traditional password systems, e.g., off-line dictionary attacks against the server password database. The establishment of unified trust infrastructure within individual organizations lays a foundation to solve other data security and individual privacy issues in this dissertation and beyond.

We were then ready to study security issues beyond organizational boundaries. Our next proposal was an anonymous remote login scheme that enables physicians or patients to access clinical services and data from off-site locations in an anonymous and unlinkable manner. This is important, as sensitive information on users such as individual preferences, life styles, health conditions is conveyed from the services (even without knowing the content) they are accessing. The proposed login scheme was robust to DoS attacks, a feature essential yet hard to achieve for anonymous systems. We believe the proposal is a useful tool when users care about user privacy in the login process.

In nature, the anonymous login scheme deals with a relatively simple scenario, which is still at the level of individual organizations. We then studied a more complex, inter-organizational process, namely, medication prescription. Medication prescription is a routine process in health care, involving multiple parties and individual privacy having distinct implications with respect to different parties. We clarified and addressed these privacy concerns by proposing a smart card enabled electronic medication prescription system. Smart card was extensively used as both a portable repository carrying up-to-date personal medical and insurance information, and a computing device for electronically signing prescription pads. To make the system more accord with real world practice, we implemented delegation of signing, a feature that enables patients to delegate their prescription signing capabilities to other people, e.g., their guardians. We proposed a strong proxy signature scheme to implement the functionality of delegation of signing.

Our final proposal continued the kind of study on “achieving user privacy while enabling medical research” as in the medication prescription system, but considered a quite different scenario: a health care organization outsources the health data in its autonomous database to other organizations, which actually involves "secondary" use of health data for, e.g., research purposes. In such cases, the health data to be outsourced are an aggregation of medical records rather than individual records, and the outsourcing care organization does not have direct business association with the receiving organizations with respect to the data to be outsourced. Privacy protection therefore should be enforced upon beyond individual data items, and the outsourcing organization has more interests to be protected against the receiving organizations. We recognized two important protection objectives: protection of individual privacy referred to in the data, as well as copyright enforcement over the data. We presented a unified framework that seamlessly combines techniques of binning and digital watermarking to attain the dual goals of privacy and copyright protection. Our binning method is built upon an earlier approach of generalization and suppression by allowing a broader concept of generalization, and our watermarking algorithm watermarks the binned data in a hierarchical manner by exploiting the very nature of the data. We implemented the techniques and obtained promising experimental results.

Some of the techniques we proposed in this dissertation may need more efficient alternatives in some situations. For example, it has been shown that ensuring k-anonymity in general is NP-hard [137]; while our proposed binning algorithm in Chapter 6 does not have efficiency problem if we deal with relatively fewer quasi-identifying columns, it will not be the case when the quasi-identifying columns are large in number; we therefor need to find more efficient privacy enhancing techniques working with digital watermarking to achieve similar protection objectives. Therefore, improvement of some of the proposals in this dissertation is included in our future agenda.

Implementing or incorporating our proposals into practical health care systems at the application level is clearly one of our main future focus. To that end, we need to consider (1) efficient enforcement of our technical proposals upon the access policies of the care organizations; (2) effective adaption of the proposals to the underlying data and relevant health standards such as HL7 [83] and DICOM [58].

Another direction for our future work is to develop health care application with *provable security*. Information security in general is quite peculiar, in the sense that we should not only construct a system, but also *make it secure*. Provable security provides a *proof* that a system is secure in the theoretic sense. A common approach for provable security is to define the desired security objectives by means of probability theory, and further demonstrates that the underlying system can meet the anticipated purposes, provided that some well-accepted computational assumptions (e.g., factorization) hold. Considering the nature of health care applications, the confidence of provable security is clearly a desirable objective to ensue.

Bibliography

[1] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik, *A Practical and Provably Secure Coalition-Resistant Group Signature Scheme*, Proc. Advances in Cryptology, Crypto'00, LNCS 1880, pp. 255-270, 2000.

[2] G. Ateniese, R. Curtmola, B. D. Medeiros, and D. Davis, *Medical Information Privacy Assurance: Cryptographic and System Aspects*, Proc. 3rd Conference on Security in Communication Networks SCN'02, 2002.

[3] ActiveX for Healthcare (AHC), Microsoft Healthcare Users Group.

[4] http://csrc.nist.gov/CryptoToolkit/aes/.

[5] D. G. Amblard, *Query-preserving Watermarking of Relational Databases and XML documents.* Proc. PODS, pp. 191-201, 2003.

[6] J. G. Anderson, *Clearing the Way for Physicians' Use of Clinical Systems, Communications of the ACM*, Vol. 40, No 8, pp. 83-90, 1997.

[7] R. Agrawal, and J. Kiernan, *Watermarking Relational Databases*, Proc. VLDB, VLDB, pp.155-166, 2002.

[8] G. Ateniese, and B. D. Medeiros, *Anonymous E-Prescriptions*, Proc. ACM Workshop on Privacy in the Electronic Society WPES'02, 2002.

[9] S. Aljareh, and N. Rossiter, *Towards Security in Multi-agency Clinical Information Services*, Health Informatics Journal, 08(02), 2002.

[10] R. J. Anderson, *Information technology in medical practice: safety and privacy lessons from the United Kingdom*, Australian Medical Journal.

[11] R. J. Anderson, *A Security Police Model for Clinical Information Systems*, IEEE Symposium on Security and Privacy, pp. 30-45, 1996.

[12] R. J. Anderson, *Security in Clinical Information Systems*, BMA consultation document, 1996.

[13] R. J. Anderson, *Problems with the NHS Cryptography Strategy*, 1997.

[14] R. Agrawal, and R. Srikant, *Fast algorithms for mining association rules*, Proc. International Conference on Very Large Data Bases, pp. 12-15, 1994.

[15] T. Albert, *Doctors Ask AMA to Assure Some Privacy for Their Prescription Pads*, http://www.ama-assn.org/sci-pubs/amnews/pick_00/prl11225.htm, 2000.

[16] http://www.atmel.com/

[17] J. Biskup, and G. Bleumer, *Reflections on Security of Database and Datatransfer Systems in Health Care*, IFIP Congress (2), pp. 549-556, 1994.

[18] B. Barber, *The Protection of Individuals by Protecting Medical Data in EHRs*, Electronic Health Records and Communication for Better Health Care, Proc. EuroRec '01, pp.38-43, 2002.

[19] D. B. Barker, M. Barnhart, T. T. Buss *PCASSO: Applying and Extending State-of-the-Art Security in the Healthcare Domian*, Proc. Annual Computer Security Application Conference, pp. 251-260, 1997.

[20] E. Bresson, O. Chevassut, and D. Pointcheval, "Security Proofs for an Efficient Password-Based Key Exchange, ACM. Computer and Communication Security, pp. 241-250, 2003.

[21] D. Boneh, *The Decision Diffie-Hellman Problem*, 3rd International Algorithmic Number Theory Symposium, LNCS 1423, pp. 48-63, 1998.

[22] F. Bao, and Robert H. Deng, *Privacy Protection for Transactions of Digital Goods*, Proc. International Conference on Information and Communications Security, LNCS 2229, pp. 202-213, Springer-Verlag, 2001.

[23] G. Bleumer, *Cryptographic Mechanisms for Health Care IT-Systems*, in (B. Barber etc., edi.) Towards Security in Medical Telematics: Legal and Technical Aspects, SHTI Vol 27, IOS-Press, pp. 233-237, 1996.

[24] O. Bukhres, and D. Hoang, *CORBAR-Based Architecture for Image Workflow in a Large Consortium of Hospitals*, International Symposium on Distributed Objects and Applications, 1999, pp. 252-263.

[25] *Biometrics*, A Journal of the International Biometric Society, http://tibs.org/biometrics/

[26] J. Barkley, *Application Engineering in Health Care*, Internal Report, Computer Systems Laboratories NIST, 1995, http://hissa.nist.gov/rbac/proj/paper/paper.html.

[27] J. Brainard, A. Juels, B. Kaliski, and M. Szydlo, *A New Two-Server Approach for Authentication with Short Secret*, Proc. USENIX Security, 2003.

[28] W. Boebert, and R. Kain, *A Practical Alternative to Hierarchical Integrity Policies*, Proc. 8th National Computer Security Conference, 1985.

[29] K. Beznosov, *Requirement for Access Control: US Healthcare Domain*, Proc. 3rd ACM Symp. Access control models and technologies, pp. 43.

[30] K. Beaver, *Information Security Issues that Healthcare Management must Understand*, Journal of Healthcare Information Management, Vol. 17, No. 1, pp. 46-49, 2003.

[31] S. Bellovin, and M. Merritt, *Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks*, IEEE Symposium on Research in Security and Privacy, pp. 72-84, 1992.

[32] S. Bellovin, and M. Merritt, *Augmented Encrypted Key Exchange: A Password-Based Protocol Secure Against Dictionary Attacks and Password File Compromise*, ACM. Computer and Communication Security, pp. 244-250, 1993.

[33] D. B. Baker, D. R. Masys, R. L. Jones, and R. M. Barnhar, *Assurance: the power behind PCASSO security*, 1999 Annual Symposium of the American Medical Informatics Association.

[34] M. K. Boyarsky, *Public-key Cryptography and Password Protocols: The Multi-User Case*, ACM Conference on Computer and Communication Security, pp. 63-72, 1999.

[35] B. Blobel, and P. Pharow, *Security Infrastructure of an Oncological Network Using Health Professional Cards*, Health Cards '97, Series in Health Technology and Informatics,Vol. 49, IOS Press Amsterdam, pp.323-334, 1997.

[36] B. Blobel, P. Pharow, K. Engel, V. Spiegel, and R. Krohn, *Communication Security in Open Health Care Networks*, Proc. Medical Informatics Europe'99, pp. 291-296, 1999.

[37] M. Bellare, D. Pointcheval, and P. Rogaway, *Authenticated Key Exchange Secure Against Dictionary Attacks*, Advance in cryptology, Eurocrypt'00, pp. 139-155, 2000.

[38] M. Bellare, P. Rogaway, *Random Oracles are Practical: A Paradigm for Designing Efficient Protocols*, ACM. Computer and Communication Security, pp. 62-73, 1993.

[39] B. Blobel, and F. F. Roger, *A Systematic Approach for Secure Health Information Systems*, International Journal of Medical Informatics, 2000.

[40] S. A. Buckvich, H. E. Rippen, and M. J. Rozen, *Driving Towards Guiding Principles: A Goal for Privacy, Confidentiality, and Security of Health Information*, Jorunal of the American Medical Informatics Association, Vol 6(2), pp. 122-133, 1999.

[41] I. Cox, J. Boom, and M. Miller, *Digital Watermarking*, Morgan Kaufmann, 2001.

[42] E. Coiera, and R. Clarkee, *e-Consent: The Design and Implementation of Consumer Consent Mechanisms in an Electronic Environment*, Jouernal of the American Medical Informatics Association, Vol 11, pp. 129-140, 2004.

[43] K. J. Cios, Krzysztof J. Cios, and J. Kacprzyk, *Medical Data Minning and Knowledge Discovery*, National Academy Press, Springer Verlag, 2001.

[44] D. Chaum, *Untraceable Electronic Mail Return Addresses, and Digital Pseudonyms*, Communications of teh ACM, Vol. 24(2), pp. 84-88, 1981.

[45] D. Chaum, *Security Without Identification: Transaction Systems to Make Big Brother Obsolete*, Communications of the ACM, 28(10), pp. 1030-1044, 1985.

[46] CEN TC 251 prENV 13729: *Health Informatics - Secure User Identification - Strong Authentication using Microprocessor Cards (SEC-ID/CARDS)*, 1999.

[47] D. Chaum, E. van Heyst, *Group Signatures*, Proc. Advances in Cryptology, Eurocrypt'91, LNCS 547, pp. 257-265, 1991.

[48] J. Camenisch, and E. Van Herreweghen, *Design and Implementation of the idemix Anonymous Credential System*, ACM. Computer and Communication Security, pp. 21 - 30, 2002.

[49] Group 4, Cancer Imaging Informatics Workshops, *Access to Databases - Security, Confidentiality, Onwership, Integrity*, 2002.

[50] K. Cole, *HIPAA Compliance: Role Based Access Control Model*, http://www.giac.org/practical/Kenneth_Cole_GSEC.doc

[51] CORBAmed: OMG's Healthcare domain task force.

[52] the Commision of the European Communities DG XIII/F AIM, *Data Protection and Confidentiality in Health Informatics*, IOS Press, 1991.

[53] Committee on Maintaining Privacy and Security in Health Care Applications of the National Information Infrastructure, *For the Record: Protecting Electronic Health Information*, National Academy Press, Washington, D.C., 1997.

[54] R. Chandramouli, *A Framework for Multiple Authorization Types in a Healthcare Application System*, 17th Annual Computer Security Applications Conference (ACSAC), 2001.

[55] T. S. Chan, *Integrating Smart Card Access to Web-Based Medical Information System*, ACM Symposium on Applied Computing, pp. 246-250, 2003.

[56] W. Diffie, and M. Hellman. *New Directions In Cryptography*, IEEE Transactions on Information Theory, IT No.2(6), pp.644C654, 1976.

[57] Department of Health, UK, *Report on the Review of Patient-identifiable Information*, 1997.

[58] DICOM, http://medical.nema.org/.

[59] J. C. Dennis, *Privacy and Confidentiality of Health Information*, Jossey-Bass, A Wiley Company, SanFrancisco, 2001.

[60] V. S. Dimitrov, G. A. Jullien, and W. C. Miller, *Complexity and fast algorithms for multi-exponentiations*, IEEE Transactions on Computers, vol 49, no 2, pp. 141C147, 2000.

[61] G. Duncan, and D. Lambert, *Disclosure-limited Data Diessemination*, Journal of the American Statistical Association, 81(393), pp. 10-28, 1986.

[62] J. D. Ferrer, J. M. Sanz, and V. Torra, *Comparing SDC Methods for Microdata on the Basis of Information Loss and Disclosure Risk*, Proc. of NTTS and ETK, 2001.

[63] D. Pointcheval, *The Composite Discrete Logarithm and Secure Authentication*, Proc. PKC'00, LNCS 1751, Springer-Verlag, pp.113-128, 2000.

[64] D. Domingos, A. Rito-Silva, and P. Veiga, *Authorization and Access Control in Adaptive Workflows*, Proc. 8th European Symposium on Research in Computer Security, ESORICS'03, LNCS 2808, pp. 23-38, 2003.

[65] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen, *SPKI Certificate Theory*, the Internet Engineering Task Request for Comments (IEFT RFC) 2693, 1999.

[66] D. Fillingham, *Exploration of the Use of Partition Rule Based Access Control (PRBAC) for Medical Applications*, http://www.anassoc.com/PRBAC

[67] Food and Drugs Administration, *Medwatch: The FDA Safety Information and Adverse Event Reporting Program*, http://www.fda.gov/medwatch/.

[68] T. A. Ferris, G. M Garrison, and H. J. Lowe, *A Proposed Key Escrow System for Secure Patient Information Disclosure in Biomedical Research Databases*, Proc. AMIA Annual Symposium, pp. 245-249, 2002.

[69] W. Ford, and B. S. Kaliski Jr, *Sever-assisted Generation of a Strong Secret from a Password*, IEEE. 9th International Workshop on Enabling Technologies, 2000.

[70] A. O. Freier, P. Karlton, and P. C. Kocher, *Secure Socket Layer 3.0*, internet Draft. 1996.

[71] D. Ferraiolo, and R. Luhn, *Role-based Access Controls*, Proc.15th NIST-NCSC National Computer Security Conference, pp. 554-563, 1992.

[72] S. E. Fienberg, *Statistical Perspectives on Confidentiality and Data Access in Public Health*, Stat Med, 20(9-10), pp. 1347-1356, 2001.

[73] J. Fox, R. Thomson, *Clinical Decision Support Systems: A Discussion of Quality, Safty and Legal Liability Issues*, Proc. AMIA Annual Symposium, pp. 265-269, 2002.

[74] B. Glicksman, Y. alSafadi, *Objects in Healthcare - focus on standards*, ACM Standards View '98, 1998.

[75] E. Gabber, P. Gibbon, Y. Matias, and A. Mayer, *How to Make Personalized Web Browsing Simple, Secure, and Anonymous*, Proc. Financial Cryptography, FC'97, pp. 17-31, 1997.

[76] L. Gong, M. Lomas, R. Needham, and J. Saltzer, *Protecting Poorly Chosen Secrets from Guessing Attacks*, IEEE Journal on Seclected Areas in Communications, 11(5), pp. 648-656, 1993.

[77] C. Georgiadis, I. Mavridis, G. Pangalos, and R.Thomas, *Flexible Team-based Access Control Using Contexts*, Proc. 6th ACM Symposium on Access Control Models and Technologies, 2001.

[78] O. Goldreich, *Secure Multi-party Computation*, Working Draft, Version 1.3, June 2001.

[79] http://www.gprd.com

[80] I. R. Greenshields, and Y. Zhihong, *Framework for Security Analysis and Access Control in a Distributed Service Medical Imaging Network*, IFIP International Information Security Conference, pp.391-400, 2000.

[81] Health Care Financing Administration, *Study of Pharmaceutical Benefit Management*, http://www.hcfa.gov/research/pharmbm.pdf, 2001.

[82] http://www.healthsmartcard.net/.

[83] HL7, http://www.hl7.org/.

[84] HL7 XML Special Interest Group.

[85] M. Hashiba et al, em Accessing Endoscopes Images for Remote Conference and Diagnosis Using WWW Server with a Secure Socket Layer, Journal of medical systems, Vol. 24, No. 6, 2000, pp. 333-338.

[86] Office for Civil Rights, *National standards to protect the privacy of personal health information*, http://www.hhs.gov/ocr/hipaa/

[87] D.S. Johnson et al, *Transferring Medical Images on the World Wide Web for Emergency Clinical Management: A Case Report*, BMJ 316 (7136):988, March 28, 1998.

[88] S. Halevi, and H. Krawczyk, *Public-key Cryptography and Password Protocols*, ACM. Computer and Communication Security, pp. 122-131, 1998.

[89] HPC (1999), *The German HPC Specification for An Electronic Doctor's Licence*, Version 0.81, Feb. 1999, http://www.hpc-protocol.de.

[90] S. F. Hbner, *A Formal Task-based Privacy Model and its Implementation: An Updated Report*, Proc. 2nd Nordic Workshop on Secure Computer NORDSEC'97, 1997.

[91] A. Hundepool, and L. Willenborg, *μ- and τ- argus: Software for Statistical Disclosure Control*. Proc. 3r International Semiar on Statistical Confidentiality, 1996.

[92] http://www.infineon.com/

[93] ISHTAR Consortium, *Implementing Secure Healthcare Telematics Applications in Europe*, Studies in Health Technology and Informatics, Vol. 66, IOS Press, 2001.

[94] ISO/IEC 7816-4:1995, Information technology - Identification cards - Integrated circuit(s) cards with contacts - Part 4: Interindustry commands for Interchange.

[95] ISO/IEC 7816-8:1999, Information technology - Identification cards - Integrated circuit(s) cards with contacts - Part 8: Security related interindustry commands.

[96] ISO/IEC 7816-9:2000, Information technology - Identification cards - Integrated circuit(s) cards with contacts - Part 9: Additional interindustry commands and security attributes.

[97] V. S. Iyengar, *Transforming Data to Satisfy Privacy Constraints*, Proc. SIGKDD, pp.279-288, 2002.

[98] *Bill to Protect Personal Data*, Japan, 1999.

[99] M. Jurecic, and H. Bunz, *Exchange of Patient Records - Prototype Implementation of a Security Attributes Service in X.500.* ACM Conference on Computer and Communications Security 1994, pp. 30-38.

[100] N. F. Johnson, Z. Duric, and S. Jajodia, *Information Hidding: Steganography and Watermarking - Attacks and Countermeasures*, Kluwer Academic Publishers, 2000.

[101] D. P. Jablon, *Password Authentication Using Multiple Servers*, RSA Security Conference, LNCS 2020, pp. 344-360, 2001.

[102] K. Jinman, D. F. Dagan, T. C. Weidong, and E. Stefan, *Integrated Multimedia Medical Data Agent in E-Health*, VIP2001.

[103] V. Jagannathan, Y. V. Reddy, and S. Friedman, *Secure Software Components for Healthcare Enterprises*, http://www.careflow.com/docs/SecureSoft.htm.

[104] H. Jepsen, *IT in Healthcare: Progress Report*, IEEE Computer Society, 2003.

[105] K. H. Kluge, *the Ethics of Electronic Patient Records*, Peter Lang, ISBN 0-8204-5259, 2001.

[106] N. Keene, W. Hobbie, and K. Ruccione, *Childhood Cancer Survivors: A Practical Guide to Your Future*, O'Reilly & Associates Inc., 2000.

[107] J. Kohl, C. Neuman, *the Kerberos Authentication Service (v5)*, internet RFC 1510. 1993.

[108] *Act for the Protection of Personal Information Maintained by Public Agencies*, South Korea, 1994.

[109] J. Katz, R. Ostrovsky, and M. Yung, *Efficient Password-Authenticated Key Exchange Using Human-Memorable Passwords*, Advances in Cryptology, Eurocrypt'01, LNCS 2045, pp. 475-494, 2001.

[110] S. Kim, S. Park, and D. Won, *Proxy Signatures, Revisited*, Proc. International Conference on Information and Communication Security, ICICS'97, LNCS 1334, pp. 223-232, 1997.

[111] D. Kalra, P. Singleton, D. Ingram, J. Milan, J. MacKay, D. Detmer, and A. L. Rector, *Security and Confidentiality Approach for the Clinical E-Science Framework (CLEF)*, Proc. 2nd UK E-Science "All Hands Meetings", pp825-832, 2003

[112] J. Katz, R. Ostrovsky, and M. Yung, "Efficient Password-Authenticated Key Exchange Using Human-emorable Passwords, Advances in Cryptology, Eurocrypt'01, LNCS 2045, pp.475-494, 2001.

[113] J. Katz, R. Ostrovsky, and M. Yung, *Forward Secrecy in Password-Only Key Exchange Protocols*, Proc. Security in Communication Networks, 2002.

[114] J. Kim, and W. Winkler, *Masking Microdata Files*, ASA (American Statistical Association) Proc. on Survey Research Methods, pp. 114-119, 1995.

[115] W. B. Lee, and C. C. Chang, *User Identification and Key Distribution Maintaining Anonymity for Distributed Computer Network*, Comput Syst Sci Eng: 15(4), pp. 113-116, 2000.

[116] D. Lambert, *Measures of Disclosure Risk and Harm*, Journal of Official Statistics, 9(2), pp. 313-331, 1993.

[117] C. Lambrinoudakis, and S. Gritzalis, *Managing Medical and Insurance Information Through a Smart-Card-Based Information System*, Journal of Medical Systems, Vol.24, No.4, pp. 213-234, 2000.

[118] Z. Lin, M. Hewett, and R. B. Altman, *Using Binning to Maintain Confidentiality of Medical Data*, American Medical Informatics Association Annual Symposium, pp. 454-459, 2002.

[119] N. Y. Lee, T. Hwang, and C. H. Wang, *On Zhang's Nonrepudiable Proxy Signature Schemes*, Pro. 3rd Australasian Conference on Information Security and Privacy, ACISP'98, pp. 415-422, 1998.

[120] J. Ledbetter, *Is Buying Drugs on the Web too Easy?* http://www.cnn.com/TECH/computing/9906/29/drugs.idg/index.html, 1999.

[121] B. Lee, H. Kim, and K. Kim, *Strong Proxy Signature and Its Applications*, Proc. SCIS, pp. 603-608, 2001.

[122] D. F. Linowes, and R. C. Spencer, *How Empolyers Handle Employees' personal Information*, http://www.kentlaw.edu/ilw/erepj/v1n1/lino-main.htm, 1997.

[123] Y. J. Li, V. Swarup, and S. Jajodia, *Constructing a Virtual Primary Key for Fingerprinting Relational Data*, Proc. ACM Workshop on Digital Rights Management, pp. 133-141, 2003.

[124] D. R. Masys, and D. B. Baker, *Patient-Centered Access to Secure Systems Online (PCASSO): A Secure Approach to Clinical Data Access Via the World Wide Web*, Annual Fall Symposium of the American Medical Informatics Association, 1997.

[125] D. Masys, D. Baker, and A. Butros, and K. E. Cowles, *Giving Patients Access to Their Medical Records via the Internet: The PCASSO Experience*, Journal of the American Medical Informatics Association, 2002.

[126] *Medwatch: The FDA safety information and adverse event reporting program*, Food and Drugs Administration, http://www.fda.gov/medwatch/.

[127] MEDSEC Consortium, *Security Standards for Health Care Information Systems*, IOS Press, 2002.

[128] M. Girault, *An Identiyy-based Identificaiton Scheme based on Discrete Logarithms Modulo A Composite Number*, Proc. Eurocrypt'90, pp. 481-486, Springer-Verlag, 1991.

[129] I. Mavridis, C. Georgiadis, G. Pangalos, and M. Khair, *Access Control based on Attribute Certificates for Medical Intranet Applications*, J. Medical Internet Research, Vol 3, Iss 1, 2001.

[130] I. Mavridis, G. Pangalos, and M. Khair, *eMEDAC: Role-based Access Control Supporting Discretionary and Mandatory Features*, Proc. 13th IFIP Working Conference on Database Security, 1999.

[131] I. Maveridis, G. Pangalos, M. Khair, and L. Bozios, *Defining Access Control Mechanisms for Privacy Protection in Distributed Medical Databases*, Proc. IFIP Working Conference on User Identification and Privacy Protection,1999.

[132] G. Bleumer, and M. Schunter. *Privacy Oriented Clearing for the German Health Care System.* in Ross Anderson (ed.): Personal Information Security, Engineering and Ethics, Springer-Verlag, pp. 175-194, 1997.

[133] B. Malin, and L. Sweeney, *Determining the Identifiability of DNA Database Entries*, Proc. AMIA Symp, pp. 537-541, 2000.

[134] P. Mackenzie, T. Shrimpton, and M. Jakobsson, *Threshold Password-Authenticated Key Exchange*, Advances in Cryptology, Crypto'02, LNCS 2442, pp. 385-400, 2002.

[135] M. Mambo, K. Usuda, and E. Okamoto, *Proxy Signature for Delegating Signing Operation*, Proc. 3rd ACM Conference on Computer and Comminications Security, 1996.

[136] V. Matyáś Jr.,*Protecting Doctor's Identity in Drug Prescription Analysis*, Health Informatics Journal, 4.4, 1998.

[137] A. Meyerson, and R. Williams, *General k-anonymization is hard*, Technical Report 03-113, Carnegie Mellon School of Computer Science, 2003.

[138] National Assoc. Health Data Organizations, *A Guide to State-Level Ambulatory Care Data Collection Activities*, 1996.

[139] B. C. Neuman, *Proxy-based Authorization and Accounting for Distributed Systems*, Proc. 13th International Conference on Distributed Computing Systems, pp. 283-291, 1993.

[140] R. Neame, *Smart cardsCthe key to trustworthy health information systems*, BMJ. 314: pp.573-577, 1997.

[141] *Privacy and Confidentiality: Access Control In Healthcare Information Systems.* http://www.careflow.com/docs/whitepaper/AccessControl.htm

[142] J. S. Park, K. P. Costello, T. M. Neven, and J. A. Diosomito, *A Composite RBAC Approach for Large, Complex Organizations*, Proc. 9th ACM Symposium on Access Control Models and Technologies, pp. 163-172, 2004.

[143] H. Pertersen, and P. Horster, *Sefl-certified Keys-Concepts and Applications*, Proc. Communications and Multimedia Security, IFIP, pp. 102-116, 1997.

[144] the European Union Privacy Directive 95/46/EC.

[145] http://www.healthprivacy.org.

[146] S. Pellissier, *Effective Authentication in a Medical Environment*, 17th Annual Conference & Exhibition, TEPR 01', 2001.

[147] Council of Europe, *On the Protection of Medical Data, Recommendation R(75)*, February, 1997.

[148] J. E Ries, P. V. Asaro, A. Guillen, and J. Ivanova, *the Futility of Common Firewall Policies: An Experimental Demonstraton*, Proc. AMIA Annual Symposium, 2000.

[149] J. Reid, I. Cheong, M. Henricksen, and J. Smith, *A Novel Use of RBAC to Protect Privacy in Distributed Health Care Information Systems*, Australasian Conference on Information Security and Privacy, ACISP 2003, LNCS 2727, pp. 403-415, 2003.

[150] M. D. Raimondo, and R. Gennaro, *Provably Secure Threshold Password-Authenticated Key Exchange*, Advances in Cryptology, Eurocrypt'03, LNCS 2656, pp. 507-523, 2003.

[151] S. Rohrig, and K. Knorr, *Towards a Secure Web-Based Health Care Application*, Proc. of the European Conference on Information Systems ECIS 2000.

[152] K. Raina, *PKI Security Solution for the Enterprise: Solving HIPAA, E-Paper Act, and Other Compliance Issues*, Wiley Publishing, Inc., 2004.

[153] R. Rivest, A. Shamir, and L. Adleman, *A Method for Obtainning Digital Signature and Public-key Cryptosystem*, Commun. ACM, NO. 21(2), pp. 120-126, 1979.

[154] A. Shamir, *Identity-based cryptosystems and signature schemes*, Advances in Cryptology, Crypto '84, LNCS196, pp. 47-53, 1984.

[155] R. Sion, M. Atallab, and S. Prabhakar, *On Watermarking Numeric Sets*. Proc. IWDW, LNCS 2613, pp. 130-146, 2002.

[156] R. Sion, M. Atallah, and S. Prabhakar, *Rights Protection for Relational Data*, Proc. SIGMOD 2003, 98-109.

[157] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd Edition, Wiley publishing, 1995.

[158] R. Sandu, M. Bellare, and R. Ganesan, *Password Enabled PKI: Virtual Smartcards vs. Virtual Soft Tokens*, 1st Annual PKI Research Workshop, pp. 892-96, 2002.

[159] C. Schnorr, *Efficient Identification and Signature for Smart Cards*, Advances in Cryptology, CRYPTO'89, LNCS 435, pp. 235-251, 1989.

[160] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and E. E. Youman, *Role-based Access Control Models*, IEEE Computer, pp. 38-47, 1996.

[161] the SEISMED Consortium, *Data Security for Health Care, Volume I: Management Guidelines*, IOS Press, 1996.

[162] the SEISMED Consortium, *Data Security for Health Care, Volume II: Technical Guidelines*, IOS Press, 1996.

[163] the SEISMED Consortium, *Data Security for Health Care, Volume III: Users Guidelines*, IOS Press, 1996.

[164] Security Glossary, http://www.garlic.com/ lynn/secgloss.htm.

[165] H. M. Sun, and B. T. Hsieh, *On the Security of Some Proxy Signature Schemes*, Cryptology ePrint Archive, NO. 068, 2003.

[166] H. E. Smith, *A Context-Based Access Control Model for HIPAA Privacy and Security Compliance*, 2001.

[167] Singapore Medical Association, *Medical Ethics & Health Law*, http://www.sma.org.sg/cmep/.

[168] L. Sweeney, *Datafly: A System for Providing Anonymity in Medical Data*, Proc. Database Security, pp. 356-381, 1998.

[169] S. Craver, N. Memon, B. Yeo, and M. Yeung, *Can Invisible Watermarks Resolve Rightful Ownerships?* Technique Report RC 20509, IBM Research Division, 1996.

[170] P. Samarati, *Protecting Respondents' Identities in Microdata Release*, IEEE Trans. Knowledge Engineering, 13(6), pp. 1010-1027, 2001.

[171] R. Sion, *Proving Ownership over Categorical Data*, Proc. ICDE, 2004.

[172] P. Samarati, and L. Sweeney, *Protecting Privacy when Disclosing Information: K-Anonymity and Its Enforcement Through Generalization and Suppression*, Technical Report, SRI International, 1998.

[173] B. Schneier, and A. Shostack, *Breaking Up Is Hard to Do: Modeling Security Threats for Smart Cards*, Proc. USENIX Workshop on Smart Card Technology, pp. 175-185, 1999.

[174] MRI 5th Annual Survey of EHR Trends and Usage. Medical Records Institute, USA, 2003. http://www.medrecinst.com/uploadedFiles/resources/survey/

[175] J. Starren, S. Sengupta, G. Hripcsak, G. Ring, R. Klerer, and S. Shea, *Making Grandma's Data Secure: A Security Architecture for Home Telemedicine*, Proc. AMIA Annual Symposium, 2001.

[176] H. Subramaniam, and Z. Q. Yang, *Report on DIMACS Working Group on Privacy/Confidentiality of Health Data*, http://dimacs.rutgers.edu/SpecialYears/2003_CSIP/reports.html.

[177] S. Tzelepi, D. Koukopoulos, and G. Pangalos, *A Flexible Content and Context-based Access Control Model for Multimedia Medical Image Database Systems*, Proc. 8th ACM Workshop on Multimedia and Security, WMS'01, pp. 52-55, 2001.

[178] S. Tzelepi, and G. Pangalos, *A flexible Role-based Access Control Model for Multimedia Medical Image Database Systems*, Information Security Conference, ISC 2001, LNCS 2200, pp. 225-346.

[179] T. C. Ting, *Privacy and Confidentiality in Healthcare Delivery Information System*, Proc. 12th IEEE Symp. Computer-Based Medical Systems.

[180] http://www.rmis.com/db/agencyihealt.php.

[181] V. Varadharajan, P. Allen, and S. Black, *An Analysis of teh Proxy Problem in Distributed Systems*, Proc. IEEE Symposium on Research in Security and Privacy, pp. 255-275, 1991.

[182] T. S. Wu, and C. L. Hsu, *Efficient User Identification Scheme with Key Distribution Preserving Anonymity for Distributed Computer Networks*, Computer & Security: 23(2), pp. 120-125, 2004.

[183] M. Wunderlich, A. Ott, J. Bernauer, and M. Leichsenring, *MEDIANOVO - A Medical Database for Medical Education, Research and Health Care*, Proc. AMIA Annual Symposium, pp. 1080, 2003.

[184] L. Willenborg, and T. D. Waal, *Statistical Disclosure Control in Practice*, Lecture Notes in Statistics, Vol. 111, Springer-Verlag, 1996.

[185] W. G. Wang, *Team-and-Role-Based Organizatinal Context and Access Control for Cooperative Hypermedia Environments*, Proc. 10th ACM Conference on Hypertext and hypermedia, pp. 37-46, 1999.

[186] ITU-T, REC. *X.509 the Directory - Authentication Framework*, 1993.

[187] Y. J. Yang, F. Bao, and R. H. Deng, *A New Architecture for Authentication and Key Exchange Using Password for Federated Enterprises*, to appear in 20th IFIP International Information Security Conference, SEC'05, 2005.

[188] W. Yancey, W. Winkler, and R. Creecy, *Disclore Risk Assessment in Perturbative Microdata Protection*, Technical Report 2002-01, Statistical Research Division, Bureau of the Census.

[189] E. J Yoon, K. Y Yoo, *Cryptanalysis of Two User Identification Schemes with Key Distribution Preserving Anonymity*, Proc. 7th International Conference on Information and Communications Security (ICICS 2005), LNCS 3783, pp. 315 - 322, 2005.

[190] L. Zhang, G. J. Ahn, and B. T. Chu, *A Role-based Delegation Framework for Healthcare Information Systems*, Proc. 7th ACM symposium on Access control models and technologies, SACMA'02, pp. 125-134, 2002.

[191] K. Zhang, *Threshold Proxy Signature Schemes*, Pro. Information SecurityWorkshop, Japan, pp. 191-197, 1997.