Enforcing Privacy via Access Control and Data Perturbation

A thesis submitted for the degree of
Doctor of Philosophy
Jian Zhong B.E., M.Sc,
School of Computer Science and Information Technology,
Science, Engineering, and Technology Portfolio,
RMIT University,
Melbourne, Victoria, Australia.
30th August, 2013
Declaration

I certify that except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; and, any editorial work, paid or unpaid, carried out by a third party is acknowledged.

Jian Zhong

School of Computer Science and Information Technology

RMIT University

26th August, 2013
Acknowledgments

This is one of the best moments in my doctoral program - to publicly acknowledge those who have contributed, in many different ways, to make my success a part of their own.

First of all, I would like to express my gratitude to my supervisor Associate Professor Peter Bertok, my second supervisor Professor James Harland, and acknowledge their contributions into this work as well as into my research career. Peter taught me how to think creatively and how to become a good researcher. Without him, this work would not have been possible. Thanks are due to the RMIT Computer Science and IT department administration staff and other academic staff, for their encouragement and intellectual inspiration. This research was partially supported by the Australian Research Council Grant No. LP0989756.

Furthermore, it is a pleasure to express my thanks to Professor Lynn Batten and Kalpana Singh, who helped summarizing attack methods for my work. Thanks are also due to Dr. Vinod Mirchandani, who helped proofreading papers and part of the thesis. Attack methods described in the Appendix were adapted to my solutions by Kalpana Singh based on [135], and I performed the calculations for the results
presented in this thesis. I would also like to thank my officemate Shaahin Madani for the help with using some research tools.

I want to acknowledge my special thanks to my mother Jingmei LI and father Lihong ZHONG for bringing me up and for their lifelong love. They are the peaceful harbor for me whenever I feel tired and frustrated. I would also like to express my thanks to Tong LIN, the most beautiful girl in the world, for her continued encouragement, caring, sharing and love.

30th August 2013
Credits

Portions of the material in this thesis have previously appeared in the following publications:


J. Zhong, V. Mirchandani and P. Bertok, "Identity Protection against Data Linkage in mHealth", In proceeding of Advanced Technology of Information Security (ATIS 2010), Melbourne, Victoria, Australia

J. Zhong, P. Bertók and Z. Tari, "Security, Privacy and Interoperability in Heterogeneous Systems". In proceeding of 11th IFIP WG 5.5 Working Conference on Virtual Enterprises, PRO-VE 2010, St. Etienne, France, October 11-13, 2010


This work was partially supported by the Australian Research Council and the Distributed Systems and Networking group at RMIT University.
# Contents

Abstract.......................................................................................................................... 1

1. Introduction.................................................................................................................. 4
   1.1 Overview and Motivation....................................................................................... 6
      1.1.1 Privacy Aware Data Access Control............................................................ 6
      1.1.2 Privacy Protection of Published Data............................................................ 14
   1.2 Contributions.......................................................................................................... 20
   1.3 Structure of the Thesis........................................................................................... 23

Part I Privacy Preserving Access Control................................................................. 27

2. Access Control in Cross-Domain Environments................................................. 29
   2.1 Introduction........................................................................................................... 30
      2.1.1 Chapter outline............................................................................................. 32
   2.2 Literature Review................................................................................................. 33
      2.2.1 Key Concepts............................................................................................... 33
      2.2.1.1 Privacy-Aware Access Control............................................................... 33
      2.1.2 Granular Data Access Control.................................................................... 34
      2.2.2 Previous Solutions....................................................................................... 35
      2.2.2.1 Diverse Privileges.................................................................................... 35
      2.2.2.2 Cross-domain Application.................................................................... 37
   2.3 Privacy Preserving Data Access Control Model (PPDAC)............................... 39
2.3.1 Basic Concepts and Notation.................................................................40

2.4 Diverse Privilege Controller of PPDAC..................................................47
  2.4.1 Privacy Preserving Subject Privilege Control (PSPC) Component.......49
  2.4.2 Privacy Preserving Object Control (PPOC) Component..................50
  2.4.3 Privilege Refinement (PR) Component............................................53
  2.4.4 Subject Roaming and Object Roaming............................................56

2.5 Illustrating Examples and Implementation.........................................62
  2.5.1 General Example...............................................................................62
  2.5.2 Specific Examples.............................................................................64
  2.5.3 Implementation..................................................................................69

2.6 Discussion.............................................................................................71
  2.6.1 Comparison with Existing Solutions.................................................71
  2.6.2 Evaluation..........................................................................................73

2.7 Summary..................................................................................................74

3. Purpose-based PPDAC.............................................................................76
  3.1 Introduction.............................................................................................76
    3.1.1 Chapter outline................................................................................78
  3.2 Background.............................................................................................79
  3.3 Privilege-oriented Purpose-based PPDAC.........................................83
    3.3.1 Basic Concepts and Notation.............................................................84
  3.4. Privilege-Oriented Purpose-Based Module........................................88
    3.4.1 Subject-based Access Control (SBAC).............................................89
      3.4.1.1 Subject Attributes Assignment (SAA) Layer..............................91
      3.4.1.2 Subject Privilege Interpretation (SPI) Layer..............................92
      3.4.1.3 Subject Privacy Label Generation (SPLG) Layer.......................96
3.4.2 Object-Based Access Control (OBAC)......................................................... 97
3.4.2.1 Object Attribute Assignment (OAA) Layer........................................... 98
3.4.2.2 Object Privilege Interpretation (OPI) Layer......................................... 100
3.4.2.3 Object Privacy Label Generation (OPLG) Layer................................. 102
3.4.3 Privilege Refinement (PR)........................................................................ 103

3.5 Model Verification......................................................................................... 106
3.6 Illustrating Example..................................................................................... 110
3.7 Discussion...................................................................................................... 116
3.8 Summary........................................................................................................ 117

Part II Privacy Preserving for Published Data................................................. 119

4. Data Privacy Protection Framework for

   Data Publishing.............................................................................................. 121

   4.1 Introduction................................................................................................. 121

   4.1.1 Chapter Outline....................................................................................... 124

   4.2 Background............................................................................................... 124

   4.2.1 Concepts and Notation........................................................................... 124

   4.2.2 Generalization......................................................................................... 126

   4.2.3 Anatomization and Permutation............................................................ 128

   4.2.4 Perturbation............................................................................................ 129

   4.3 Data Privacy Protection Framework......................................................... 130
5. Chebyshev Data Perturbation ....................................................... 133

5.1 Introduction.................................................................................. 133

5.2 Mathematical Foundations - Chebyshev Polynomials.................. 135

5.3 Proposed Method - Chebyshev Data Perturbation (CDP)............ 137

5.3.1 Overall Process Flow.............................................................. 137

5.3.2 The Proposed Chebyshev Perturbation Algorithm................... 138

5.3.2.1 Calculating the Perturbation Values.................................... 139

5.3.2.2 Perturbation................................................................. 144

5.3.2.3 Restoration................................................................. 146

5.4 Experiments and Results............................................................ 147

5.4.1 Evaluation and Experimental Setup........................................ 147

5.4.1.1 Distribution Tests....................................................... 148

5.4.1.2 Empirical Information Content Tests................................ 149

5.4.2 Experimental Performance.................................................... 150

5.4.2.1 Distribution Tests....................................................... 151

5.4.2.2 Information Content.................................................... 155

5.4.2.3 Attack Resistance....................................................... 157

5.5 Discussion.................................................................................... 157

5.6 Summary...................................................................................... 159

6. μ-Fractal Data Perturbation ....................................................... 161

6.1 Introduction.................................................................................. 161

6.2 Mathematical Foundations........................................................ 163

6.3 μ-Fractal Data Perturbation (μ-FDP)........................................ 167
6.3.1 Overview of μ-FDP................................................................................ 168
6.3.2 Perturbation Algorithm........................................................................... 171
   6.3.2.1 Initial Parameters and Fractal Sequences........................................ 172
   6.3.2.2 Perturbation Vector 1................................................................. 173
   6.3.2.3 Perturbation Vector 2................................................................. 174
   6.3.2.4 Perturbation................................................................................ 175
6.3.3 Restoration.................................................................................................. 176
6.4 Experiments and Results............................................................................... 176
   6.4.1 Evaluation and Environmental Setup.................................................. 177
      6.4.1.1 Distribution................................................................................ 177
      6.4.1.2 Information Content................................................................. 177
   6.4.2 Experimental Performance.................................................................... 178
      6.4.2.1 Distribution Tests....................................................................... 178
      6.4.2.2 Empirical Information Content Tests......................................... 181
      6.4.2.3 Attack Resistance Tests............................................................ 183
6.5 Discussion.................................................................................................... 183
6.6 Summary...................................................................................................... 185

7. Conclusion and Future Work......................................................................... 185

Appendix A

Appendix B

Bibliography
List of Figures

Figure 1.1: Typical Access Control System.................................................................6
Figure 1.2: Thesis Structure........................................................................................25
Figure 2.1: P-RBAC Family Model [39]. .................................................................34
Figure 2.2: Data Granularity.......................................................................................34
Figure 2.3: Hospital Role Mapping Table [211]. .......................................................37
Figure 2.4: Privacy Preserving Data Access Control Model (PPDAC)....................39
Figure 2.5: Duty and Roles.......................................................................................43
Figure 2.6: Hierarchical PSPC..................................................................................46
Figure 2.7: Overall Process Flow of PPDAC on Granular Privilege.......................48
Figure 2.8: PSPC Process..........................................................................................49
Figure 2.9: Core PPOC.............................................................................................51
Figure 2.10 Dynamic Hierarchy with HA and CA.....................................................52
Figure 2.11: Privilege Refinement.............................................................................54
Figure 2.12: Data Roaming.......................................................................................58
Figure 2.13: Object Roaming ....................................................................................79
Figure 2.14: Subject Roaming...................................................................................60
Figure 2.15: Dual Roaming Request..........................................................................61
Figure 2.16: Module and Component Diagram for The Implemented Scenario........63
Figure 2.17: A Scenario for A Hospital System.......................................................64
Figure 2.18: Illustrating Example.............................................................................65
Figure 2.19: Subject Grades......................................................................................66
Figure 6.10: Distribution Test -- Normal Distribution with PA ≈ 6.8% .......................... 178
Figure 6.11: Distribution Test -- Uniform Distribution with PA ≈ 6.8% .......................... 179
Figure 6.12: Distribution Test -- Normal Distribution with PA ≈ 12.5% .......................... 179
Figure 6.13: Distribution Test -- Uniform Distribution with PA ≈ 12.5% .......................... 180
Figure 6.14: Distribution Test -- Normal Distribution with PA ≈ 22.4% .......................... 180
Figure 6.14: Distribution Test -- Uniform Distribution with PA ≈ 22.4% .......................... 181
Figure 6.12: Information Content Test with PA≈6.8% ................................................. 182
Figure 6.13: Information Content Test with PA≈12.5% ................................................. 182
Figure 6.14: Information Content Tests with PA≈22.4% ................................................. 183
List of Tables

Table 2.1: Features Comparison ........................................................................................................... 72
Table 3.1: Sample Activities Mapping Table .......................................................................................... 95
Table 4.1: Example of format for micro-tables ....................................................................................... 125
Table 5.1: Comparison of the Methods .................................................................................................... 158
Table 6.1: Summarized Notion ................................................................................................................. 171
Abstract

With the increasing availability of large collections of personal and sensitive information to a wide range of user communities, services should take more responsibility for data privacy when disseminating information, which requires data sharing control. In most cases, data are stored in a repository at the site of the domain server, which takes full responsibility for their management. The data can be provided to known recipients, or published without restriction on recipients. To ensure that such data is used without breaching privacy, proper access control models and privacy protection methods are needed.

This thesis presents an approach to protect personal and sensitive information that is stored on one or more data servers. There are three main privacy requirements that need to be considered when designing a system for privacy-preserving data access. The first requirement is privacy-aware access control. In traditional privacy-aware contexts, built-in conditions or granular access control are used to assign user privileges at a fine-grained level. Very frequently, users and their privileges are diverse. Hence, it is necessary to deploy proper access control on both subject and object servers that impose the conditions on carrying out user operations. This thesis
defines a dual privacy-aware access control model, consisting of a subject server that manages user privileges and an object server that deals with granular data. Both servers extract user operations and server conditions from the original requests and convert them to privacy labels that contain access control attributes. In cross-domain cases, traditional solutions adopt roaming tables to support multiple-domain access. However, building roaming tables for all domains is costly and maintaining these tables can become an issue. Furthermore, when roaming occurs, the party responsible for multi-domain data management has to be clearly identified. In this thesis, a roaming adjustment mechanism is presented for both subject and object servers. By defining such a dual server control model and request process flow, the responsibility for data administration can be properly managed.

The second requirement is the consideration of access purpose, namely why the subject requests access to the object and how the subject is going to use the object. The existing solutions overlook the different interpretations of purposes in distinct domains. This thesis proposes a privilege-oriented, purpose-based method that enhances the privacy-aware access control model mentioned in the previous paragraph. It includes a component that interprets the subject's intention and the conditions imposed by the servers on operations; and a component that caters for object types and object owner's intention.

The third requirement is maintaining data utility while protecting privacy when data are shared without restriction on recipients. Most existing approaches achieve a high level of privacy at the expense of data usability. To the best of our knowledge, there is no solution that is able to keep both. This thesis combines data privacy protection
with data utility by building a framework that defines a privacy protection process flow. It also includes two data privacy protection algorithms that are based on Chebyshev polynomials and fractal sequences, respectively. Experiments show that the both algorithms are resistant to two main data privacy attacks, but with little loss of accuracy.
Chapter 1

Introduction

We live in the information age, and personal information has become one of the most valuable resources. This includes personal data (e.g. date of birth, address, age, gender etc.), personal preference data (e.g. habits and hobbies) and generated personal data (e.g. medical data) [142]. Commercial data consumers can use these data for service or product improvement to target specific customers or to reduce market costs, and such information is also widely used for research, data modeling and government statistics. Various data repositories store significant amount of personal data, which include statistics bureaux, healthcare data centers, education institutes, non-profit organizations and companies. To make the best use of the stored data, it is usually released in one form or another, commercially or non-commercially.
CHAPTER 1 INTRODUCTION

Often, data contain sensitive or confidential information. Releasing data without any precaution (e.g. without proper sanitization) can lead to privacy problems, such as improper use of data or the release of personally identifiable information. In fact, privacy issues have been a major concern in the utilization of personal data [1]. To eliminate the threat of privacy breaches and to comply with various organizational policies, legal regulations, subscription conditions and so forth, when data is shared with data consumers, privacy preserving techniques have to be implemented [157].

Privacy preserving techniques can be classified into two categories, based on the target data consumers, namely data sharing with known recipients and data sharing with unknown recipients. The former indicates the data consumers are known and traceable by the data holder, whereas the latter indicates the data are free for any parties to use and these parties usually cannot be traced or data access cannot be further controlled after the data has been published.

Privacy protection techniques implemented in the first category are usually called privacy aware data access control. These techniques regulate the data consumers’ privileges so as to prevent improper user behavior in relation to the received data. Privacy protection techniques in the second category are usually called data privacy
CHAPTER 1 INTRODUCTION

protection or privacy preserving for data publishing. These two categories will be
discussed in Part I and Part II of the thesis, respectively.

1.1 Overview and Motivation

1.1.1 Privacy Aware Data Access Control

Data access control is one of the fundamental information management mechanisms.
It determines the availability of resources, permissible user behavior and deals with
related conditions [2]. Figure 1.1 depicts a typical access control system. The user
represents the data consumer who requests data access. A policy-based decision
maker interacts with users and when a user requests access to a resource, it forms an
appropriate access request that includes the necessary control attributes. Then, such a
request is passed to the system and applied according to the system policies. Lastly,
the decision maker permits or denies the user’s request.

![Diagram of Access Control System]

*Figure 1.1: Typical Access Control System*
CHAPTER 1 INTRODUCTION

Based on the general model above, there are different approaches in this category with different focuses. Some may protect information stored in a database, or information relating to the method or parameters of access.

A common method of requesting access is via the web, such as using web services, social networks etc. Web-based services, such as email or location based services (LBS), are receiving more and more attention. One problem is selective revealing the contents to an authorized party, showing certain parts while hiding other, sensitive content. The approach in [3] uses a pre-analysis component to split private information from the original data, to avoid any privacy breach. In other cases, the users’ privacy has to be protected when essential user information is accessed by the parties who provide services for the users (e.g. in case of LBS near by shops, gas station, restaurant etc.). Existing work includes the study of user privacy concern [4], evaluation of risk [5, 61, 62], balancing submitted information and received services, installation of middleware [6], grouping to hide individuals [7], adding dummy data [8] and perturbing the path [9]. Privacy aware access control for social networks is a new topic and the main problems are mostly related to user (data owner) behavior and how the service providers use the sensitive data. Research on this topic includes survey [10, 11] and evaluation of threats [12, 13] and reducing accuracy in order to get better privacy [14]. Works also include re-assigning identity with minimal
collusion [63-65]. Privacy aware web-based policy enforcement mainly works on standard computer-readable formats for privacy policies and protocols that enable web browsers to read and process privacy policies automatically, such as P3P [15] and its improvements that include enterprise-oriented solutions [16] and how to make use of P3P [17].

In case of databases, in addition to protection against unauthorized access, data may also need to be protected from the database service provider. The main problems are keeping the privacy of the queries sent by the user, privacy of the data requested and minimizing the workload at the user front-end and maximizing the efficiency on the server side. Iyer et al. [18, 19] proposed a solution for minimizing the client workload, based on a graphical representation of queries as trees. Hacigumus et al [20] show a method for splitting the equivalent query into two sub-queries so that one of them satisfies the user’s request and the other may insert irrelevant tuples (decided by privacy/security methods). Methods for query protected data also include a bucket-based approach [21] that sorts non-overlapping subsets of values dividing into buckets of the same, predetermined size. An improved bucket-based method [22] proposed an efficient way for partitioning the domain of attributes by minimizing the number of spurious tuples in the result of both range and equality queries. However, bucket-based methods are vulnerable to inference attacks [138]. A hash-based
CHAPTER 1 INTRODUCTION

approach [23] generates the hash of attributes instead of using plaintext but does not support range queries. This limitation is considered by adopting B+ trees [23], which is an indexing method allowing every vertex to store up to \( n-1 \) search key values and \( n \) pointers and, except for the root and leaf vertices, has at least \( n/2 \) children. Some solutions reduce the amount of irrelevant tuples [26, 27] by applying a secure hash function to each pair of subsequent characters of each index value, e.g. index value \( s \) has \( n \) characters \( c_1c_2..c_n \) and the corresponding index is \( \text{hash}(c_1c_2)\text{hash}(c_3c_4)\ldots\text{hash}(c_{n-1}c_n) \). Other approaches consider encrypted databases that employ homomorphic encryption to allow query operations to be executed over index values [20, 28, 29]. Since not all data are considered private in a database, only the sensitive ones need protection. An approach [19] only encrypts sensitive attributes and leaves others in plaintext; then searching keywords in encrypted documents [30-33] will produce all documents containing a particular keyword without the need to know any other information. Another approach [34] modeled privacy requirements through confidentiality constraints such as sets of attributes and approaches to enforce privacy policies [35, 36].

In organizational scenarios, data access control usually concerns user privilege control, user privilege in collaboration environment and purpose-based access control. Part I of the thesis focuses on this category because it is significant for contemporary
data sharing scenarios and improper access can lead to a privacy leak. The focuses in these scenarios are as following [100, 101, 118].

- Unique users
- Complex (diverse) privileges, containing various privileges and conditional privileges
- Fine-grained data control
- Cross-domain application
- Purpose-based

Unique users and complex privileges are often considered together in the literature [100, 101]. A particularly difficult issue is administering unique users who come and leave system and have diverse privileges that include not only simple operations, such as *read* an object, but also complex ones (e.g. *edit*, *sign*) and can have associated conditions, such as *if...then*.... In Role-Based Access Control (RBAC) privileges are assigned to users via role subscriptions: privileges are assigned to roles and each user is classified into one or more roles. In such a model, users are not able to acquire permissions directly, only through their roles, which simplifies many common operations, such as adding a user, or changing a user's department [37, 38].
While RBAC is often used, unique users with diverse privileges are very hard to assign to traditional roles [101]. A number attempts have been made to support unique user management [102-105] and complex privileges [107-108]. To support unique users in RBAC, roles can have attributes [49, 110] or parameters that tailor a role to the individual user [129, 42]. Approaches include different conditional role approaches, such as user access conditions [39, 49, 129], user component-based [97] and situation-based [98] access control.

Other important issues are providing access to parts of data items, referred to as fine-grained, granular access control. Mechanisms have been proposed for this in [107, 108] and approaches considering both granular data and user privilege control have been presented in [102-105, 108, 110]. User-oriented and data-oriented management have to be considered together for complete access control.

As more and more parties are working together and sharing data among each others, the control mechanisms for a single domain have to be extended for cross-domain applications, and that involves user privilege and data access control adjustments. These adjustments are to support subject (user) and object (data) roaming. Subject roaming denotes that users temporarily join a foreign domain and request access to resources in the user domain while object roaming denotes that an object is requested
by a foreign subject and such object needs to be delivered to such foreign domain. Different domains may have different roles, resources and access rules, and roaming user role adjustment and privilege refinement can be difficult [44]. Solutions usually build a roaming table to map a role in one domain to a role in another domain [111-112], while others adopt user agents [113-115]. For the cross-domain environments, many attempts are providing roaming adjustment support, such as roaming table [111-112], attribute mapping [46], temporarily assigning constraints [47], user agents [113-114], policy agents [115], threats detection [43, 45] and multi-domain relationship approach [44].

However, the existing solutions are not able to cater for diverse privileges with conditions on fine-grained granular data. They also overlooked server constraints either on the user or on the data side. Furthermore, none of them are able to fully satisfy the needs of user roaming, data roaming and both happen at the same time. Therefore, a new user privilege control model is needed for roles with diverse privileges, and to allow many unique users come and leave in cross-domain environments.

The main problems of building the new user privilege control model are i) a large number of users need different complex privileges, which makes it hard to group all
users into roles; ii) collaboration control on granular data and iii) user roles and privileges maintenance in cross-domain environments. Once users and resources are roaming across multiple domains, the management of responsibility becomes complex, such as who should be responsible for a roaming user requesting access to roaming data. The problems stated above are addressed by the proposed dual control model containing user privilege and granular data, with a roaming adjustment mechanism (Chapter 2).

Although the problem of unique users with diverse privileges in cross-domain environments has been looked at, for privacy preserving access control the purpose of access also need to be considered [118]. To address this problem, purpose-based access control (PBAC) has emerged that regulates access according to purpose. A classic PBAC model, such as [118], uses access purpose as the basis of access control. This was later improved with the definition of intended purpose [116], developing an organizational model [50] and the introduction of usage control [126]. Other extensions include the support of conditional roles [127], conditional intended purpose [125], spatial role and spatial purpose [44], intended purpose management [117, 122], privilege chain [119] and purpose flow management [120, 123]. In addition, a data element-oriented model was proposed in [124]. These models improve the applicability of PBAC to practical scenarios. However, these purposes
depend heavily on users, application domains and environments. The meaning of a purpose can be translated to different operations and may lead to distinct privileges in cross-domain environments. Ignoring this can cause privilege conflicts in collaborating scenarios.

The main problem that needs to be investigated in purpose-based access control is cross-domain purpose translation and privilege adjustment. This will be addressed by the proposed purpose based access control model (Chapter 3).

The problems of data publishing to known recipients have a significant impact on many privacy-aware access control models. Models that solve these problems are able to provide privacy protection for data sharing to known recipients, provide better flexibility for large enterprises, and enable total management for data in cross-domain environments. Lack of concerns of the problems can cause privacy breach and affect operation performance during data sharing.

1.1.2 Privacy Protection of Published Data

As there is a loss of control over data once published, in such scenarios data require additional protection, such as anonymization (or de-identification [48]), to avoid any
privacy or security breach [51]. Data anonymization is widely adopted by the census bureau, healthcare data centers and government agencies [57].

Traditional data anonymity approaches simply removed identifying fields from the released data, such as social security number and name. However, some studies have shown that even without any personally identifiable information (PII) [138], a collection of certain personal attributes can still enable the identification of a large proportion of the population. In an example described by Sweeney [52], a dataset collected by an insurance commission contained medical records of Massachusetts state employees. Although any identifiers such as name, social security numbers and phone numbers were removed from the data, a large number of individuals were still identified by using information such as date of birth, post code and gender from a voter registration list. As it turned out, among those identified was the state governor who authorized the data release [52]. Another research [53] showed that around 87% of the population of the United States can be uniquely identified using the seemingly innocuous attributes of gender, date of birth and 5-digit zip code. In another case, AOL published a 2 GB file containing approximately 20 million search queries from 650,000 of its users and the anonymization scheme used to protect the data consisted of assigning a pseudonym random number to each AOL user and replacing the user ID with this number [55]. Later on, two New York Times reporters used the search
key words such as name of the town, last name, age-related information etc to re-identify a few persons from the published data [57]. Netflix, a movie rental service, announced the Netflix Prize for the development of an accurate movie recommendation algorithm based on a large amount of movie rating information for 18,000 movie titles [58]. Soon Frankowski et al [59] pointed out the potential risk and then the amount of rating data was successfully attacked [60].

The information used to identify individuals in the above three examples is called Quasi-identifier (QI). Clearly, whether a piece of information can be a QI depends on its usability to identify individuals rather than on the data type. The principle behind the identity revealing process is that by linking several released data sets, the overlap of the data sets becomes smaller until the overlap can uniquely identify individuals.

This identity revealing process is also called data linkage or triangulation attack.

The three main approaches to providing data privacy are generalization and suppression, anatomization and permutation, and perturbation [73].

*Generalization and suppression* is one way of resisting data linkage attacks. It is exemplified by $k$-anonymity [53,66] that ensures that for every record in the released data table, there are at least $k-1$ other records that have exactly the same values for the
quasi-identifiers. This can be achieved by data suppression or generalization for example. However, several limitations of the $k$-anonymity were found later [68] and an improved method called $l$-diversity was proposed [67]. The $l$-diversity model requires that every group of indistinguishable records contains at least one distinct sensitive attribute value. Later, several privacy models were proposed, that quantified adversarial knowledge such as (c,k)-safety [69] limited the maximum privacy disclosure to less than $c$, and 3D privacy criterion [70] for safe data release. These models are stricter than $k$-anonymity but are hard to implement in real life as the data can be random, and satisfaction of these models can require even more modification than $k$-anonymity does. Also, there are some extensions to $k$-anonymity and $l$-diversity such as [72] presented a model by combining randomization and data transformation, but it is yet to be realized. The full domain generalization method, such as [74], generalizes attributes to the same value, while in the subtree generalization approach [75-78] at a nonleaf node (attribute), either all child values or none are generalized. Such generalization method is also called global re-coding. At the same time, cell generalization [79-80], also called local re-coding, allows some values of an attribute remain un-generalized. For example, let us have two classes called “professionals” and “artists”. “Professionals” contains “engineers” and “lawyers”, and “artists” contains “writers” and “dancers”. Global re-coding happens if every “lawyer” is generalized to “professional”, as well as every “engineer” to
“professional”. Local re-coding happens if one “engineer” is generalized to “professional”, while allowing other “engineers” to remain unchanged. An extended generalization method, called multi-dimentional generalization considers multiple QI attributes as a tuple, and each QI can be decided whether to be generalized regardless of other QI attributes [81-83]. There are also different suppression schemes. Value suppression [84-85] refers to suppressing every value of a given attribute in a dataset, while cell suppression [86], also called local suppression, refers to suppressing some values of a given attribute in a dataset.

Anatomization and permutation: Anatomization de-associates QIs and sensitive attributes rather than modifying either of them, such as in [87]. Permutation partitions a set of data records into groups and shuffles their sensitive values within each group [88].

Perturbation is similar to generalization as it also modifies the data, but has the advantage of being reversible if an accessor has the restoration key. A data linkage attack cannot be performed as the real data is not available to unauthorized users [96]. Perturbation works with additive or multiplicative noise [89-92, 149, 152], data swapping [79, 87, 93, 150, 167] and/or synthetic data generation [94, 95].
Although generalization and suppression cannot totally eliminate the threat of data linkage attacks, the principle of changing the target values and making them indistinguishable from the original effectively reduces the possibility of such attacks [73]. Data privacy is preserved if the adversaries are not able to derive the original data from the modified (e.g. perturbed) data or the re-constructed results are not close enough to the original data.

By adopting the above privacy preservation definition, most generalization techniques are able to meet this requirement. However, such techniques pay a high cost in data utility; generalization changes the distribution and other data features of the data partly or totally. Anatomization and permutation are vulnerable to data linkage attacks, while perturbed data changes the data format or range, and keeps only certain data statistical properties [73].

The central problem of privacy preserving data publishing is how to keep data privacy while maintaining data utility. Depending on the application, utility can be data distribution, data format and data range, and an authorized data recipient should be able to restore the original data while others can only access processed data [48].
1.2 Contributions

There are two main research questions that will be investigated in Part I and Part II of the thesis:

i) How to preserve privacy of data shared with known users?

ii) How to protect privacy of published data while maintaining data utility?

In Part I, the thesis proposes a privacy preserving data access control (PPDAC) model that can be instantiated for practical applications. The model is built according to the information privacy protection concept in [51] and it considers three aspects: users (subjects), sensitive data (objects) and controlled disclosure (access privileges). In addition, the model extends the features of the privacy aware role-based access control model (P-RBAC) and the purpose-based access control model (PBAC) by creating a multi-layer control model that deals with collaboration management of both users and resources. The model focuses on user privileges and builds a solid, controllable, label-based mechanism to handle granular data.

Part II of the thesis proposes data perturbation algorithms to protect the privacy of published data and maintain data utility. The algorithms generate perturbation noise and combine it with the original data in order to thwart data reconstruction attacks.
CHAPTER 1 INTRODUCTION

[73]. In addition, a controllable perturbation mechanism ensures the processed data’s utility.

The contributions of this thesis are as follows.

When sharing data with known recipients, the proposed model:

1) Provides flexible user privilege control that
   - Allows large number of unique users be handled;
   - Controls user privileges in combination with user roles and user attributes, so that in an organization the same role can have different privileges;
   - Supports a hierarchical user attributes model for accessing fine-grained data;

2) Supports complex diverse user privileges, by:
   - Enabling compound user privileges that contain diverse privileges and privilege conditions;
   - Allowing for complex user behavior management and complex data access tasks via user operation sequence control;
   - Enabling multiple conditions for user access, which can contain logical expressions.
CHAPTER 1 INTRODUCTION

3) Catering for granular data control, to assist:

- Individual control for each fine-grained granular data item;
- Hierarchical granular data attribute control incorporating a user hierarchical attribute model to form a dual control model
- Handling multiple data attributes, data categories and category specifications

4) Includes purpose-based control that considers:

- User access purpose, user obligations and server constraints
- Data purpose containing data type, allowed purpose and prohibited purpose, and data server constraints.
- Hierarchical purpose to handle fine-grained data access requests

5) Supports cross-domain applications via:

- User role and attributes adjustment and
- data attribute adjustment when roaming across domains;
- Dynamic purpose translations between domains;
- Identifying responsibility of access enforcement when dispute occurs

Systems that use the proposed approach can implement privacy protection for data shared between known recipients.
CHAPTER 1 INTRODUCTION

When data is shared with unknown recipients, the proposed methods:

6) Allow the restoration of the original data by authorized users

7) Keep the data

- In the same format as the original data;
- In the same range as the original data;
- Indistinguishable from the original data;
- Distribution close to the original data (when the original data follows normal
or uniform distribution)

8) Resists different attacks, in particular:

- Data linkage attacks;
- Spectral Filter (SPF) attacks;
- Bayes-Estimated Data Reconstruction (BE-DR) attacks.

9) Increase entropy more than $k$-anonymity and $l$-diversity in most cases.

10) Allow the control of perturbation magnitude to meet different needs.

1.3 Structure of the Thesis

As illustrated in Figure 1.2, the thesis consists of seven chapters in two parts.
CHAPTER 1 INTRODUCTION

The first part addresses the issue of data sharing with known recipients. Chapter 2 presents the first functional module of the proposed Privacy Preserving Data Access Control (PPDAC) model, supporting unique users with diverse privileges in cross-domain environments. Chapter 3 presents the other module of PPDAC model for cross-domain purpose translation, purpose representation, purpose management including multiple purposes for users (user access purposes, server constraints) and handling granular data with hierarchy data type attributes, allowed purposes, prohibited purposes and data server constraints.

Part II addresses the issue of data sharing with unknown recipients by developing data perturbation techniques. Chapter 4 explains a data privacy protection framework (DP$^2$F) and reviews the literature. It also introduces evaluation methods. Chapters 5 and 6 present two data privacy protection algorithms that are based on Chebyshev polynomials and fractal sequences, respectively. Attack resistance is introduced and examined in the Appendix. Finally, the thesis is concluded in Chapter 7 and future work is suggested.
CHAPTER 1 INTRODUCTION

Figure 1.2: Thesis Structure
Part I

Privacy Preserving Access Control
PART I
PRIVACY PRESERVING ACCESS CONTROL

Data has become a valuable commodity, and to avoid its misuse access to it needs to be controlled. When data is meant to be shared with others, its protection is even more important. Data can be shared with known recipients such as within an organization or among controllable domains, or with unknown recipients which usually indicates data made public.

This section discusses privacy preserving data sharing with known recipients. Here, privacy indicates the prevention of improper use of data or the release of data which can be used to identify individuals. In such scenario, privacy protection is usually enforced by access control mechanisms. A widely used mechanism is role-based access control (RBAC) which assigns user privileges according to roles. Since it was not designed to protect privacy, many other improved and enhanced solutions have been proposed. However, RBAC also lacks in some other features, such as support of unique users with diverse privileges and cross-domain applications [101].

To address these issues, this section builds a privacy preserving access control model with two built-in functional modules that deal with the enforcement of privacy preserving access control for granular data in both single-domain and cross-domain environments.

Chapter 2 looks into access control of unique users with multiple privileges in cross-domain environments. Then chapter 3 presents enhanced, purpose-based access control and focuses on user purpose and data owner's intention management, and
PART I

PRIVACY PRESERVING ACCESS CONTROL

proposes a privilege-oriented purpose-based access control mechanism. The overall model will be verified in Chapter 3.
Chapter 2

Access Control in Cross-Domain Environments

This chapter details a privacy preserving access control model for diverse privileges that are constrained by access conditions and granular data. Privacy preserving role-based access control (P-RBAC) is the traditional and common way in which a third party can be prevented from accessing information by not granting certain privileges. The proposed solution further looks into diverse privilege control for cross-domain applications by implementing a dual control model, which contains a subject server and an object server. As the names suggest, the subject server focuses on user management while the object server focuses on object management. The proposed model enables user privilege control on granular data and complex user privilege
management. The proposed model is formally verified and the verification is presented in chapter 3.

2.1 Introduction

Data integration and sharing is no longer restricted to office computers and local networks but has become an integral part of our everyday lives. It is utilized on a large scale by various organizations, corporations and government agencies [201]. Access control is crucial in case of sensitive data, for example when improper access can compromise a person’s/organization’s privacy. While a security breach in case of a credit card can be addressed by cancellation and re-issuance of the card, a personal privacy breach cannot be remedied the same way. To enforce proper access control, a widely used method is Role-based Access Control (RBAC), in which a role is viewed as a set of access permissions for people performing certain tasks, such as a division manager can read and edit particular data, and a sales person can only read the same data. A major shortcoming of traditional role-based access control is that it was not designed to enforce privacy policies and barely meets privacy protection requirements [39]. To mitigate this limitation and support privacy policies and address privacy protection, an extension of RBAC termed Privacy-aware Role-based Access Control (P-RBAC), has been proposed [39].

As requirements become more stringent, access to different parts of shared data needs to be controlled separately, and for that granular privilege control can be introduced.
A solution has been proposed for granular privileges that are constrained by access conditions and granular data, such as *sign if no amendments required* and *read only certain parts if not using an office computer* [130]. Various user privileges and data granularity make the existing solutions difficult to deploy in environments that involve many unique users with diverse access rights. As these users can have many distinct privileges, it is difficult to group them into roles [201]. At the same time, data sharing between different domains becomes more and more common, but existing approaches (such as simply mapping a subject from one role in a domain to a different role in another domain [211]) are not able to cater for privilege adjustment [46]. In addition, responsibility of data management in multiple domain environments is not clearly identified either.

Considering the restrictions of existing work, this chapter addresses the problem of granular privilege access control in both single-domain and cross-domain environments. The problem has several *challenges* that were not investigated in previous research.

- Single user account: It is desirable to define a framework that allows users in different domains access a data server by using their original accounts in the home domains, and without having an account in each domain.

- Granular data control and granular privilege control: For compound, multipart data, such as used in healthcare or in collaborating organizations, fine grained access permissions, are required.
CHAPTER 2 ACCESS CONTROL IN CROSS-DOMAIN ENVIRONMENTS

- User liability: When privacy is breached, the party responsible has to be clearly identified.

To deal with these three challenges, this chapter presents a granular privilege control model, that can be employed in both single-domain and cross-domain environment. The module controls overall access permissions, permissions on granular data and granular privileges on granular data. The experiments show that in multiple user domains and data domains users with granular privilege requests are able to access different data domains without applying for new accounts. Also, once data are distributed without permission, it is possible to identify the party responsible.

The proposed solution not only encompasses the advantages of existing privacy-aware access control mechanisms, but makes granular privilege adjustable for cross-domain organizations such as in collaborating scenarios.

2.1.1 Chapter outline

The remainder of this chapter is organized as follows. Section 2.2 reviews the literature on privilege control mechanisms and multiple domain applications. Section 2.3 presents basic concepts on data access control that will be used later in the thesis. Section 2.4 proposes a privacy preserving data access control mechanism based on granular privilege and cross-domain environments. Section 2.5 presents general and specific examples, and implementation. This is followed by the discussion of the proposed mechanism and a comparison with existing solutions. The chapter is summarized in section 2.7.
2.2 Literature Review

Privacy-aware subject (user) access control and granular data access control are key concepts in user privilege assignment. In this section, existing approaches are examined from two aspects: (i) how they can handle granular privileges from unique user requests and (ii) their suitability for cross-domain application.

2.2.1 Key Concepts

2.2.1.1 Privacy-Aware Access Control

The strategy of using a formal model to represent user rights based on role assignment is called policy-based provisioning or role-based access control (RBAC) [2, 38]. In a role-based access control system privileges are assigned to users through role membership: privileges are attached to roles and each user is classified into one or more roles. The fact that users are not able to acquire permissions directly, only through their roles, simplifies many common operations, such as adding a user, or changing a user's department [37-38].

However, traditional role-based access control was not designed to enforce privacy policies or to address privacy protection requirements [39]. Privacy-Aware Role-based Access Control (P-RBAC) is a family of models that extends the traditional
RBAC model to support privacy policies by providing hierarchical and conditional access control [39]. The overall architecture is shown in Figure 2.1. The foundation is the core P-RBAC model, which defines the basic elements. Hierarchical P-RBAC introduces the notions of role hierarchy and object hierarchy. Role hierarchy describes an inheritance relationship among roles, while object hierarchy defines a partial ordering relation between different objects. There is one more component, conditional P-RBAC, that introduces permission assignment, indicating the condition under which user is granted access. In Figure 2.1, universal P-RBAC integrates the features of conditional P-RBAC and hierarchical P-RBAC.

Figure 2.1: P-RBAC Family Model [39]

2.1.2 Granular Data Access Control

Granular data refers to the fineness with which data fields are sub-divided [40]. For example, a postal address can be recorded with coarse granularity as a single field shown in Figure 2.2, or broken down into individual items in a fine grain model.
Granular Data access control is generally governed by label-based mechanisms, i.e. the functions or algorithms are executed based on the information in the labels that are attached to subjects and objects. In some cases, different label-based policies may apply to fine-grained data to satisfy different requirements [41].

2.2.2 Previous Solutions

2.2.2.1 Diverse Privileges

Several attempts have been made to support diverse privileges [202-205]. A solution, by combining role-based and granular data access control, provides a goal-driven mechanism via incorporating context information to support variable privilege requests [207]. Users submit requests to get privileges via roles and get proper information through object models. Conditions denote privacy policies that must be satisfied before a data access request can be granted. A two-phase role engineering process is used to refine proper privileges: (i) role-permission analysis produces role and permission candidates with corresponding contexts, and then (ii) role-refinement eliminates any ambiguity and redundancy from the roles and permissions [207].
Another solution uses conditions that can be added to permissions to keep the number of roles manageable and make user privilege management flexible [208]. Similarly, in [202-205] attributes are added to the traditional role-based model to realize multiple privileges in diverse unique user scenarios. These approaches’ central idea asserts that allowing access can be determined based on various attributes presented by a subject [210]. Rules specify conditions under which access is granted or denied. For example, a bank might allow access if the subject is a teller working between the hours of 7:30 am and 5:00 pm, or the subject is a supervisor or auditor working those same hours who also have management authorization. Specifically, in [202], a requester is granted access to a collection of services based on a given collection of attributes, while in [203] multiple policies are involved in refining user privileges. The method introduced in [205] uses semantic web technologies to extend attribute-based access control to both subject server and object server, and so user privilege and data granularity are considered at the same time. In [229], diverse privileges are managed by a parameterized role model. Different from previous models, this approach replaces some parts of the roles by parameterized rules, in order to meet more complex user requirements. The model in [42] combines attribute-based solutions [202] and hierarchical access control [230], and proposes a user hierarchy to cater for unique users with diverse privileges. The model in [44] uses separation of duty (SoD) in unique user access management, and it was extended in [47] with access conditions. Nevertheless, the object model in [207-208] is not sufficient to meet granular data access control requirements, due to the lack of a systematic granular data control model. In [229], role models are difficult to build before the users lodge their requests.
CHAPTER 2 ACCESS CONTROL IN CROSS-DOMAIN ENVIRONMENTS

Approaches [202-204, 208, 210] connect subject-based with object-based privilege control, but do not address granular data control satisfactorily. This is because these methods either overlook the control of different privileges on different granular data, or overlook different conditions on both subject server and object server when user requests are lodged. Although [205] explores the combination of subject and object control in one model, it overlooks unique users with diverse privileges and object granularity. The approach in [42] lacks access condition support, and the model in [44, 47] overlooks access control of granular data and user privileges associated with duties.

2.2.2.2 Cross-domain Application

A cross-domain application requires user roaming and object roaming. Subject roaming refers to a user lodging an access request through a domain other than the user’s home domain. Object roaming indicates that an object is transferred to another domain’s data server (called object server in this thesis) rather than directly to a user. For roaming scenarios, a role-to-role mapping table, also called roaming table is used in [211]. By building such a table, the method supports users operating in multiple domains at the same time. Figure 2.3 gives an example of a hospital role mapping table. However, building mapping tables for each pair of domains complicates table management. In [212], a historical role mapping table is employed in order to reduce the size of mapping tables created for the global environment. But when a user requests privileges different from his role or when the conditions of the object server the user is accessing change, maintaining historical mapping tables becomes an issue.
In [46], attribute mapping is involved to assist user roaming, but it does not solve the user privilege adjustment.

<table>
<thead>
<tr>
<th>External Role</th>
<th>External Organization</th>
<th>Local Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td><strong>Senior Medical Student</strong></td>
<td>Purdue University</td>
<td><strong>External Medical Student</strong></td>
</tr>
<tr>
<td><strong>Senior Medical Student</strong></td>
<td>Indiana University</td>
<td><strong>External Medical Student</strong></td>
</tr>
<tr>
<td><strong>Senior Investigator</strong></td>
<td>State Farm Insurance</td>
<td><strong>External Investigator</strong></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

*Figure 2.3: Hospital Role Mapping Table [211]*

Another attempt [213, 214] introduces a user agent to allow subjects moving between different domains. To gain access to a certain data in a foreign domain, the user has to activate his role in an external organization via his home subject server. But the approach only considers subjects in different domains; it does not address object roaming across domains when data is transferred between servers in a global system. Similarly, methods in [215] also adopt policy agents to address cross-domain data sharing, but they overlook diverse user privileges and data granularity. In [43, 45], the approach works on detecting privacy access threats in cross-domain environments but does not provide a proper solution for privilege control in such environments. In [44], a concept called multi-level domain relationship is defined for cross-domain applications, but the paper does not discuss user privilege adjustment in remote domains.

To summarize the literature, the key concepts of privacy-aware subject access control and granular data access control have been implemented by existing data management mechanisms, and some previous solutions have embraced these two concepts to address strict privacy requirements in different application environments.
For users with diverse privileges, previous approaches focused on formal models of user privileges, which can be used to assign roles to users based on user classification and other user attributes. Some tried to add different modules to handle diverse privileges for users [207-209, 229]. However, in such an environment, role models are difficult to build before users lodge their requests. In addition, the overhead of unique users coming and going can represent a significant load. In real-world deployments, formal role models have not scaled well, because when many users are unique, there is no significant leverage to be gained by grouping them into roles [101]. For cross-domain applications, the issue was addressed when only a limited number of users are involved in data sharing [211], but as the number of users grows and multiple domains are involved, maintaining a roaming table becomes very complex.

### 2.3 Privacy Preserving Data Access Control Model (PPDAC)

This section outlines the structure of Part I of the thesis graphically, with emphasis on a privacy preserving data access control model (PPDAC, see Figure 2.4) that lays the foundation for the model detailed in chapter 3. Additionally, basic concepts and notation are introduced.
Figure 2.4: Privacy Preserving Data Access Control Model (PPDAC)

The model shown in Figure 2.4 illustrates the two main components to realize the proposed approach, which are the subject server and the object server. Each component has two main functional modules: diverse privilege controller and purpose-based access controller. This chapter focuses on PPDAC model and the diverse privilege controller module, and the purpose-based access controller module will be detailed in the next chapter.

2.3.1 Basic Concepts and Notation

This section defines the concepts and introduces notation first appearing in this chapter; those used in the purpose-based access controller module will be defined in chapter 3. In general, the concepts and notation will be illustrated and explained where they are first used.
**Definition 2.1 (Subject):** A Subject is an active participant such as a user or an organization. A set of subjects is \( S=\{S_i \mid i=1,2,\ldots,n\} \), where \( n \) is the number of subjects in a data sharing environment.

To make it consistent, the term ‘subject’ is used instead of ‘user’ in the rest of the thesis.

**Definition 2.2 (Object):** An Object is a passive entity. A set of objects is \( O=\{O_i \mid i=1,2,\ldots,m\} \), where \( m \) is the number of objects in a data sharing environment.

Objects are the target that should be fully or partly protected, such as a patient healthcare record or a finer-grained object like blood pressure within a patient’s healthcare record.

**Definition 2.3 (Subject Activity):** A subject activity (SA) is a user operation. A set of subject activities is \( SA=\{SA_i \mid i=1,2,\ldots,k\} \), where \( k \) is the number of subject activities and \( \Omega(SA) \) denotes a subset of \( SA \).

Examples of user operations are *read, edit, comment, redistribution* and represented by \( SA_{read}, SA_{edit} \) and so on.

**Definition 2.4 (Subject Activity Sequence):** A subject activity sequence (SAS) is a container of subject activities, their relationships and order of execution.
The relationship between two activities in an SAS can be the following.

### Subject Activity Sequence Relationships

- **$SA_1 \rightarrow SA_2$** denotes that for a subject $S$, activity in $SA_2$ must be executed after activity in $SA_1$. For example, activity *comment* must follow activity *edit*:
  
  $SA_{edit} \rightarrow SA_{comment}$

- **$SA_1 \leftarrow SA_2$** denotes that for a subject $S$, activity in $SA_2$ must be executed before activity in $SA_1$. For example, activity *read* must precede activity *comment*:
  
  $SA_{comment} \leftarrow SA_{read}$

- **$SA_1 \leftrightarrow SA_2$** denotes that for a subject $S$, activity in $SA_2$ and activity in $SA_1$ are mutually exclusive, and only one of the two activities will be processed. For example, a medical diagnosis can be waiting for either to be approved or to be edited and commented.

- **$SA_1 \uparrow SA_2$** denotes that for a subject $S$, activity in $SA_2$ and activity in $SA_1$ can be processed simultaneously or in any order. For example, the object can be *read* and *redistributed* at the same time.

So the relationship between two activities in an SAS can be, for example, that subject activity *comment* has to be executed after subject activity *edit*.

**Note:** to make the notion clearer, multiple subjects are represented by $S_A$, $S_B$, $S_C$ etc.; multiple objects are represented by $O_\alpha$, $O_\beta$, $O_\gamma$ etc.; multiple subject activities are represented by $SA_1$, $SA_2$, $SA_3$ etc.
**Definition 2.5 (Duty):** Duty is a collection of a subject, an object and the subject's activities on the object. A duty $D = \{S, SAS, O, a\}$, indicates that a duty of subject $A$ requires access to object $a$ for a set of activities.

A duty can be, for example, a medical staff member needs to read and append to a patient’s healthcare record. The relationship between duties and roles is shown in Figure 2.5. The circles denote different roles and the shaded parts belong to a duty. It shows a duty not only contains roles (the roles are usually described by SAS), but also contains a subject and the target object.

![Figure 2.5: Duty and Roles](image)

**Definition 2.6 (Access Level):** Access level indicates the importance of an object or the rank (position) of a subject. In the proposed model, access levels are represented by numeric values.
**Definition 2.7 (Subject Grade):** Subject Grade (SG) is the subject's overall access level.

A subject can access an object only when its SG is equal to or greater than the object grade (definition 2.9).

**Definition 2.8 (Object Granular Data):** Object granular data (OGD) is partial data. Every object can include a set of granular data \( \{O_{\alpha GD} \mid i = 1, 2, ..., n\} \), where \( n \) is the number of granular data in object \( \alpha \).

An example of an object granular data is a paragraph in a document or a sentence in a paragraph.

**Definition 2.9 (Subject Sub-Grade):** Subject Sub-Grade (SSG) is the subject's access level for a piece of granular data of an object.

Each SG contains a set of Sub-grades (SSGs) indicating the access level to each piece of granular data of the object. Each SSG contains an SAS indicating the execution order of the subject activities.

**Rule 2.1:** If there is no SSG in a duty request, by default the SSG is set to the same value as the SG of the subject, and different SSGs different from the same SG are handled independently.
**Definition 2.10 (Object Grade):** Object Grade (OG) is the minimum access level for a subject to be able to access the object.

If $SG \geq OG$, then access to the object is authorized, although access to some object granular data (OGD) of this object may require a higher $SSG$.

**Definition 2.11 (Object Sub-Grade):** Object Sub-Grade (OSG) is the minimum access level for a subject to be able to access a piece of granular data.

Each object ($O$) can have one or more pieces of object granular data (OGD). Each OGD is optionally assigned an object sub-grade (OSG). If a piece of OGD is assigned with an OSG, access is granted to this OGD only when the SSG for the OGD is equal or greater than the OSG of the OGD. If an OGD is not assigned with an OSG, access is granted to this OGD only when both $SG$ is equal or greater than $OG$ and $SSG$ for the OGD is equal or greater than the $OG$ of this object. If there is an OGD, but no OSG is associated with, the OGD uses $OG$ as the access level. In addition, if there is no SSG for such OGD, the subject access level for the OGD is by default equal to $SG$.

**Rule 2.2:** The access level of each object granular data $OSG_i$ must be no smaller than the object grade $OG$. When a subject with $SG \geq OG$ is able to access an object, but has no permission to any granular data in the object, i.e. $SSG_i < OSG_i \ \forall i$, then access will be granted only to the general information that is not assigned an OSG.
Definition 2.12 (Negative Permission): Negative Permission (NP) defines operations that are not allowed to be executed on an object.

NP has the highest process priority in the proposed mechanism. An example of NP is No edit permission for a certain piece of object granular data (OGD) unless SSG for OGD is greater than \( \theta \) (i.e. \( \theta \) is the minimum access requirement).

Negative permissions are used only in the object privacy label which is handled by the PPOC, and will be detailed in section 2.4.2.

Definition 2.13 (Special Condition): Special Condition (SC) defines conditions that apply on an object or object granular data.

\( SP \) is processed after NP in the proposed mechanism. An example of SP is must sign if edit.

In summary, a subject can access an object only when its SG is equal to or greater than the object grade (see definition 2.9). Each SG contains a set of subject subgrades (SSGs) indicating the access levels to each piece of granular data of the object. For each piece of object granular data, a set of SAS indicating the relationship of the activities are attached. The relationship between SG, SSG and SAS is illustrated in Figure 2.6.
The first part of Figure 2.6 shows subject sub-grade 1 (SSG₁) for access to object granular data 1, where the SSG₁ is used for comparison with OSG₁, and a subject activity sequence 1 (SAS₁) containing two subject activities SA₁ and SA₂. The activity relationship is ‘processing in any order’. The other two parts indicate the activity relationships are SA₄ must be executed before SA₃, and SA₅ and SA₆ are executed mutually exclusive.

### 2.4 Diverse Privilege Controller of PPDAC

This section describes the privilege control module shown in Figure 2.4. The module is composed of three functional components: privacy preserving subject privilege control (PSPC), privacy preserving object control (PPOC), and privilege refinement (PR). The PSPC component caters for granular privilege support and roaming user
privilege adjustment, *PPOC* provides support for object conditions and special requests, and adjustment of roaming data, and the *PR* component evaluates the privileges of users against the requested objects. Apart from the above functional components, roaming processes in cross-domain environments are also considered in section 2.4.4, which helps to identify responsibility for privacy breach such as data being redistributed without authorization. Before examining the first functional component, the overall process flow of the diverse privilege controller is explained with the help of Figure 2.7.

![Figure 2.7: Overall Process Flow of PPDAC on Granular Privilege](image)

When a user submits a request, the *PSPC* module first checks the user's identity and computes the proper permissions. For instance, a user requests a medical document with *read*, *edit*, *comment* and *redistribute* privileges, but *PSPC* may ascertain that the user is only allowed to have *read* and *comment* privileges on the destination object server (explained in section 2.4.1). At the same time, the required document is processed by *PPOC*, and a set of negative permissions (*NPs*) and special requests (*SPs*) are attached according to the data owner’s preferences (further explained in section 2.4.2). Finally, the privilege refinement module evaluates the permissions and
assigns the final access rights (see section 2.4.3). When roaming into a foreign domain, a user will first connect to the home subject server, which will communicate with the foreign subject server to procure proper privilege constraints. If an object needs to be copied to a foreign domain, a dynamic hierarchy is computed on the destination object server. The roaming scenarios are discussed in section 2.4.4.

2.4.1 Privacy Preserving Subject Privilege Control (PSPC)

Component

The PSPC is explained via its three parts: PSPC process, hierarchical PSPC and subject privacy label generator. The core PSPC defines the PSPC process flow that is depicted in Figure 2.8. Hierarchical PSPC is responsible for putting forward subject granular privilege candidates that are used by a label-based privilege system for subject activity control. The subject privacy label generator encapsulates privilege candidates that are derived from hierarchical PSPC into a privacy label and sends the label to the Privilege Refinement component (See Figure 2.7).

Figure 2.8: PSPC Process
There are three main elements defined in the PSPC process: subject (S, definition 2.1), duty (D, definition 2.5), and subject privacy label (SPL) generator. A Subject is an active participant, and a typical subject is a human user. Duty is defined in definition 2.5 and expressed by \( D = \{(S, SAS, O)\} \), which composes of a subject, subject activity sequence and target objects. A subject privacy label (SPL) is a control frame containing privilege candidates. In the PPDAC module, no negative permissions (see definition 2.12) are assigned to an SPL.

The processing in PSPC starts with a subject submitting a duty request to the hierarchical PSPC, which compiles the subject grade-subgrade hierarchy for the subject and object involved, produces a list of privilege candidates, and sends it to the subject privacy label generator.

The last part of the PSPC module is the subject privacy label generator. It takes privilege candidates from the hierarchical PSPC and encapsulates them into a subject privacy label (SPL) represented by equation (2-1), where \( i \) denotes the index of required object granular data pieces and \( n \) is the number of requested object granular data. The privilege candidates correspond to the subject activities that are evaluated by the subject server. They will be used in the privilege refinement component (section 2.4.3) to calculate proper permissions.

\[
SPL = \{SG, SSGi, SSGi.SAS\} , \quad i \in n
\]  

(2-1)
### 2.4.2 Privacy Preserving Object Control (PPOC) Component

The PPOC component implements label-based control for granular data in cross-domain applications. PPOC is designed to cooperate with subject diverse privilege control over granular objects. In this section, the three parts of PPOC are explained: core PPOC, dynamic hierarchical control and object privacy label (OPL) generator. The core PPOC works on granular data and defines the process flow of PPOC (see Figure 2.9). There are three main elements defined in core PPOC: object (O, definition 2.2), dynamic hierarchy, and object privacy label (OPL) generator. The essential concepts are explained below, with some examples showing how they fit in the core PPOC.

![Figure 2.9: Core PPOC](image)

Object here indicates the requested target, such as a file, a part of a file, data sheets or several pieces of information for example date of birth, email address etc. Granular object is a relative concept that refers to finer bits of the object. For instance, compared with object ‘address: 414-418 Swanston street, Melbourne VIC 3000’, the granular ones can be ‘address: Street number: 414-418; Street name: Swanston; City: Melbourne; State: Victoria; Post code: 3000’. The dynamic hierarchy module is the
place where a requested object is assigned the privacy control codes, such as object grade \((OG)\), object sub-grade \((OSG)\), negative permissions \((NPs)\) and special conditions \((SCs)\). The \(OPL\) generator assembles all control codes into a privacy label.

The central module of \(PPOC\) is called Dynamic Hierarchy. It has two core functions: hierarchy assignment \((HA)\) and condition assignment \((CA)\). Hierarchy assignment is the process of \(OG\) and \(OSG\) assignment, and data transformation (i.e. masking or perturbing original data. Perturbation algorithms are detailed in Chapters 5, 6 and 7). Condition assignment attaches \(NPs\) and \(SCs\) to the object privacy label. Figure 2.10 illustrates an example of dynamic hierarchy with object grade, object sub-grade and condition assignment.

![Diagram of Dynamic Hierarchy with HA and CA](image)

*Figure 2.10 Dynamic Hierarchy with HA and CA*
In the example, object sub-grade 1 indicates the access requirement of the first piece of granular data. If a subject does not meet the grade required by the object, access will be denied. Both OG and OSG are assigned by hierarchy assignment. ‘No edit permission unless \( SSG_i \) is greater than 0’ is an example of an \( NP \) coming from condition assignment. There is no SP in \( OSG_i \). In \( OSG_2 \), ‘anyone who read this has to sign’ is an \( SP \) indicating a mandatory requirement for a subject who wants to read this piece of granular data.

With proper control assignments from Dynamic Hierarchy, all of these control codes are encapsulated into an object privacy label (\( OPL \)). An \( OPL \) is a carrier of \( PPOC \) control code shown in equation (2-2), where \( n \) denotes the number of required granular data pieces in an object, and the numbers \( x \) and \( y \) depend on the object's conditions.

\[
OPL = \{OG, OSG_i.\Omega(NP)_x, OSG_i.\Omega(SP)_y, \quad i \in n \} \tag{2-2}
\]

### 2.4.3 Privilege Refinement (PR) Component

After the derivation of a subject privacy label (\( SPL \)) from \( PSPC \) (section 2.4.1) and an object privacy label (\( OPL \)) from \( PPOC \) (section 2.4.2), the final proper privileges of the subject on the object are calculated in the Privilege Refinement (\( PR \)) component.
Privilege Refinement (PR) introduces a structure to assist in defining and enforcing rules of granular privilege control. It implements dual control of subject granular privileges and object granular data.

Figure 2.11 shows that after both subject and object privacy labels are produced, authorized privileges will be calculated by PR. The calculation has four steps. First a validation check of both subject and object access levels, such as SG and OG, SSG1 and OSG1, is performed. It is followed by privilege refinement that derives proper permissions for the required object. These steps are detailed below.

Step 1: Grade computing

The subject grade (SG) in the subject privacy label (SPL) is represented by SPL.SG and similarly, the object grade (OG) in the object privacy label (OPL) is OPL.OG. If
SPL.SG is equal or greater than OPL.OG, PR will continue the refinement process, otherwise access is denied.

**Step 2: Sub grade computing**

The subject sub-grade (SSG) in the SPL is represented by SPL.SSG and similarly for the object sub-grade in OPL is OPL.OSG. Each object granular data (OGD) is assigned an OSG and there is an SSG for access request. PR goes through each OGD and compares the SSG and the OSG associated with it. If the SSG is equal or greater than OSG, PR will continue processing, otherwise access to such object granular data is denied. Even if access to a piece of granular data is denied, the decision will not affect other sub-grade computing.

**Step 3: Subject activity sequence computing**

A subject activity sequence in the subject sub-grade is represented by SPL.SSG.SAS, where \( i \) refers to the index of object granular data (OGD) and \( j \) refers to an activity such as read, add, remove etc. Object negative permission (NP) and special condition (SC) are represented by OPL.OSG.NP and OPL.OSG.SC respectively, where \( i \) indicates the index of OGD. For each OGD, the activities in OPL.OSG.NP will be removed from the SPL.SSG.SAS. If the removed activity is in a sequence relationship, the activities after it will not be executed. For example, if \( SA_{add} \rightarrow SA_{sign} \) and OPL.OSG.NP is \( SA_{add} \), then the \( SA_{sign} \) will be not executed either. After processing NP, object special condition (SC) will be added to the subject activity sequence. If the subject fails to satisfy a special condition, the activities in the SC will not be executed. For example, if a subject activity sequence is \( SA_{add} \rightarrow SA_{sign} \) and the OPL.OSG.SC is
‘no add permission after 1st Feb’, then an add privilege request will be denied after that date, and in the sequence $SA_{sign}$ will not be executed either.

**Step 4: Execution of the refined privileges**

If the access to an object and object granular data ($OGD$) is granted, the execution of activities starts from the object’s general information, such as object identifier and the information that is not assigned with $OSG$, then proceeds with each $OGD$.

This section explained how the subject privacy label and object privacy label help to calculate the subject’s final proper permissions over the requested object, and the solution for single domain environments was explained. The next section introduces the scenario of cross-domain environments.

### 2.4.4 Subject Roaming and Object Roaming

This section starts with the concepts of subject roaming and object roaming. Then it explains the roaming process in both $PSPC$ and $PPOC$ components. At last, three different examples are provided to help understanding the roaming concepts. An important feature of the solution is that the responsibilities are clearly identified.

Subject roaming happens when a subject leaves his own home domain and moves to another domain where the subject does not have an account. Object roaming happens when an object is required to be duplicated on another domain’s object server. During object roaming, the subject’s home subject server is responsible for data sharing,
whereas in subject roaming, the remote (foreign) subject server is. The roaming process in PSPC and PPOC are explained in detail below.

First let us look at an example of hierarchical PSPC processing. Let us assume that subject $S$ submits a duty query $D=\{S, SAS, O\}$ to subject server $SS$, and the subject is a valid user in the system, while $O$ represents the required object. After receiving the request, the subject server establishes (i) whether the request contains subject roaming and (ii) whether the required object is stored on the home object server or on a remote object server. Assume that the subject grade of $S$ is $SG$ and it has subject activities allowed by the local subject server $SS$. If subject $S$ does not require subject roaming, the Hierarchical PSPC module will send $SG$ and the subject activities to the privilege refinement component (see section 2.4.3). If the request refers to an object that can only be accessed on a remote server, subject roaming is needed. Let us assume subject $S$, with home subject server $SS$, has no account on the remote subject server $RSS$. If the roaming request is accepted by server $RSS$, it will generate a new subject grade represented by $R(SG)$ based on $SG$ and an allowed activity set $R(SA)$ in the $RSS$ is used instead of original $SA$. Subject roaming can be represented as $SG(Foreign\ Domain) = F[SG(Local\ Domain)]$. Here function $F$ denotes a mapping function that is predetermined in each server based on the privacy level in each domain (in reality, this may require experienced administration). Here a simple mapping function is used only. Assume subject $S$ in his home domain is assigned $SG$ and in domain $RSS$, a predetermined $R(SG)$ is associated with subject $S$. Then for each subject sub-grade in the $RSS$ is calculated by $\frac{R(SSG)}{SG} = SSG \times R(SG)$. 
When an object is required by a subject in the same domain, hierarchy assignment (HA) produces the object's original OG and puts it into the OPL (see section 2.4.3), while condition assignment (CA) puts negative permissions (NP) and special conditions (SC) into the OPL. In a data roaming situation a remote server requires a copy of object O, but the original OG and OSG may not be suitable for evaluation by the remote server directly. Figure 2.12 shows the concept of HA applied adjustment. In such cases, the home object server will first assign OG (from HA) and conditions (from CA) to the object. Then the object will be sent to the remote object server, where the Dynamic Hierarchy component will map OG into R(OG). This can be represented as \( OG(\text{Foreign Domain}) = F [OG(\text{Local Domain})] \). The function \( F \) is a mapping function that is predetermined in object servers and works in a similar way to that of subject roaming. Finally, the object will be sent to the remote subject server to satisfy the user’s request. In Figure 2.12, OG associated with object O is shown with a list of OSGs that are the object sub-grades. When object O (including its granular data) is created, it is associated with an OG. For instance, the object grade of all data on an organization’s data server are determined based on the users in this organization.

Figure 2.12: Data Roaming
The following example illustrates object roaming. A researcher wants to access a medical data sample that is only available on a remote server, on which the researcher has no account. Also, the researcher’s home server has no right to create a temporary account on the remote server. If the home server is eligible to acquire and store an authorized copy on the home object server for a period of time, object roaming can take place. The object roaming process can be expressed as follows, where $n$ indicates the total number of granular data.

**In home servers:** $\{OG (home \text{ Domain}), OSG_i(home \text{ Domain}), Conditions\}$

**In remote server:** $\{OG(Foreign \text{ Domain}), OSG_i(Foreign \text{ Domain}), Conditions\}$

Where $OG (Foreign \text{ Domain}) = F [OG(Local \text{ Domain})]$

$OSG_i(Foreign \text{ Domain}) = \{F(OSG_i(home \text{ domain})) | \forall i \}$

Next, the roaming process is discussed in three typical cases: (i) local user accessing roaming objects, (ii) roaming user accessing remote objects on the home server and (iii) roaming user accessing roaming objects.

Figure 2.13 shows that a user is trying to access an object stored on a remote object server via the home server of the local user. For example, a doctor needs a document from another hospital's data server but has no permission to access it directly. The doctor has to access this file via his home hospital. Step (1) shows that the doctor sends a request for a patient’s medical record $O$ stored on a remote server. After the doctor submits his request, the home hospital makes a request for object $O$ to the remote subject server. In step (3) the request has been accepted and in step (4) object $O$ is sent to the remote subject server and is ready for roaming. Steps (5) and (6) show
that the home hospital has received a copy of object $O$ and is available for access. In this procedure, the home hospital has to be responsible for the copy of $O$.

Figure 2.13: Object Roaming.

Figure 2.14 shows the process when a user is roaming to a remote server and requires an object stored on that server. For example, a doctor is invited to another hospital and he needs to access data stored on the server of that hospital. The process of subject roaming extends the basic local request by linking the home subject server to a remote subject server i.e. in the visited hospital. The process is similar to object roaming, but this time the data remains on the remote server, it does not propagate to the doctor’s home hospital. In this procedure, the remote subject server has to be responsible for the requested object $O$. 
CHAPTER 2 ACCESS CONTROL IN CROSS-DOMAIN ENVIRONMENTS

Figure 2.14: Subject Roaming

Figure 2.15: Dual Roaming Request
Figure 2.15 shows the process when a user $S$ is roaming to a remote server $RS_A$ and requires an object stored on another remote server $RS_B$, and the data has to be accessed via server $RS_A$. For instance, a doctor is invited to a foreign hospital for a certain period of time and needs a group of medical documents that are stored at another data centre. The dual roaming request process is used in this case. Steps (1) and (2) represent the doctor’s roaming privileges calculation, and in step (3) a request is made to the remote object server $RS_B$. This step is the same as step (2) in the object roaming request process. Steps (4) and (5) represent encapsulation of the group of medical documents' conditions, while step (6) is the data roaming process which is the same as step (5) shown in Figure 2.13. Eventually, the roaming object is delivered to remote server $RS_A$. In this procedure, remote server $RS_A$ has to be responsible for the copy of $O$.

### 2.5 Illustrating Examples and Implementation

#### 2.5.1 General Example

This section presents the demonstration of the proposed diverse privilege controller of $PPDAC$ in a simple medical environment. The two scenarios described are a normal access request and a roaming request. The operational flow of the normal access request is depicted in Figure 2.16. At first, users submit requests to the $PSPC$ component via a user interface. Then, the Hierarchical $PSPC$ module calculates the allowed privileges and produces a subject privacy label ($SPL$, section 2.4.1). In the
**PPOC** component, the requested object is passed to the Dynamic Hierarchy, in which hierarchy assignment (*HA*) and condition assignment (*CA*) take place (section 2.4.2). After this, the allowed privileges and conditions are used for generating an object privacy label (*OPL*). After both *SPL* and *OPL* have been generated, the labels are passed on to the privilege refinement component to calculate the final proper privileges.

![Diagram](image_url)

*Figure 2.16: Module and Component Diagram for The Implemented Scenario*

Figure 2.17 shows how the proposed approach can be used to manage and share data in a collaborative and cross-domain environment. The figure contains the participating organizations (domains), Departments (sub domains), positions (roles) and requests.
In this example, it is assumed that all doctors, nurses and researchers are in the same medical area of specialization. It is further assumed that doctors can work both in medical and research departments, and they can work in other hospitals if needed (the latter representing subject roaming). Nurses can only work in medical departments and in their home hospitals. Researchers can be contractors or university students, and they can only work in research departments. Researchers can ‘roam’ to other hospitals’ research departments only if accompanied by their supervising doctor.

Figure 2.17: A Scenario for A Hospital System
2.5.2 Specific Examples

To help understanding of the proposed model, this section describes a medical example, including the involved parties (domains), participants (subjects and objects), code base (privilege table and roaming rules) and individual cases that illustrate how the proposed model works.

There are three parties involved and shown in the Figure 2.18. Each party is an organization, and represents an individual domain which contains its own subject server and object server. Assume that organization A is a small regional healthcare clinic, organization B is a national level healthcare center and organization C is a national healthcare data repository and research center. Users in organization A cannot directly access data stored in organization C but have to request it via organization B.
In this section, assume there are five kind of participants, which are:

- **ICT support (IS):** IT support and management for the organizations
- **Specialist (S):** Specialist for certain category of disease.
- **General practitioner (GP):** general doctors
- **Nurse (N):** staff who collect patients’ data and input data to the system but do not make any diagnosis.
- **Head-Nurse (HN):** leader of nurses in healthcare organizations

The rules for three organizations are shown in figures below.

<table>
<thead>
<tr>
<th>Subject roles</th>
<th>Subject grades</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specialist</td>
<td>SG=9</td>
</tr>
<tr>
<td>GP</td>
<td>SG=8</td>
</tr>
<tr>
<td>Head-nurse</td>
<td>SG=7</td>
</tr>
<tr>
<td>Nurse</td>
<td>SG=6</td>
</tr>
<tr>
<td>ICT support</td>
<td>SG=5</td>
</tr>
</tbody>
</table>

*Figure 2.19: Subject Grades*

<table>
<thead>
<tr>
<th>Object category</th>
<th>Object grades and sub-grades</th>
</tr>
</thead>
<tbody>
<tr>
<td>General information in a patient’s record</td>
<td>OG=5</td>
</tr>
<tr>
<td>Medical record field in a patient’s record</td>
<td>OSG=7</td>
</tr>
</tbody>
</table>

*Figure 2.20: Object Grades*

<table>
<thead>
<tr>
<th>Activity</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>read</td>
<td>Read the specified granular data</td>
</tr>
<tr>
<td>add</td>
<td>Add information to the granular data. Note: this activity does not allow to remove information</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>remove</td>
<td>Delete information from granular data. Note: this activity does not allow to add information</td>
</tr>
<tr>
<td>edit</td>
<td>Add and remove information of granular data</td>
</tr>
<tr>
<td>comment</td>
<td>Comment to granular data. Note: this activity does not allow editing original granular data.</td>
</tr>
<tr>
<td>sign</td>
<td>Sign an object or certain part of an object. This activity is used when someone approves or acknowledges something</td>
</tr>
<tr>
<td>distribute</td>
<td>Distribute to other domains’ subjects</td>
</tr>
<tr>
<td>manage</td>
<td>Manage the object</td>
</tr>
</tbody>
</table>

Figure 2.21: Activity List For The Three Organizations

<table>
<thead>
<tr>
<th>Roaming option</th>
<th>Roaming adjustment rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>DomainA.subject S roams to DomainB</td>
<td>The subject’s SG decreases by 1 when the subject is roaming to domain B</td>
</tr>
<tr>
<td>DomainC.objectS roams to DomainB</td>
<td>The object’s OG increases by 1 when objects in domain C roam to domain B</td>
</tr>
<tr>
<td></td>
<td>Negative permissions are edit, distribution, manage when objects in domain C roam to domain B.</td>
</tr>
<tr>
<td></td>
<td>Special condition is comment and sign after all activities</td>
</tr>
</tbody>
</table>

Figure 2.22: Roaming Rules

Case 1: Assume DomainB.subject is a specialist, who wants to update a patient’s medical record object O that is stored in domain B as well. The specialist sends a duty request to subject server B which is \(D=\{\text{DomainB.subject}, \text{edit}, \text{DomainB.object}\} \).
The subject server checks and confirms that the DomainB.subject is a specialist and sets the DomainB.Subject.SG to 9.

The object server has set the DomainB.object.OG to 5 and the medical record field of the patient’s record to 7 according to the system rules. In addition, the object server requires confirmation by the person updating the medical record, which is represented as DomainB.object.SP=sign. As the medical record can not be deleted, DomainB.object.NP = remove.

The subject label is SPL = \{SG = 9, SSG=9, SSG.SA_{edit}\} and the object label is OPL = \{OG=5, OSG=7, OSG.NP = remove, OSG.SP=sign\}. The privilege refinement process is:

i) Compare SG and OG. Because SG is greater than OG, the subject is allowed to access the record’s general information, such as the record’s identity (Note: not the patient’s identity).

ii) Compare SSG and OSG. Because no specific SSG is assigned to the specialist, then the assigned SSG is equal to the value of SG. As SSG is greater than OSG, access is granted and privileges need to be refined.

iii) The subject request is SSG.SA_{edit} (updating the record), which means the specialist requests edit privilege for the granular data medical record. As the OSG.NP is remove, it means no delete privilege is granted for this data item. Thus, the remaining privilege is add because edit contains add and remove.
iv) The object contains \( OSG.SP = sign \), which means a confirmation of the specialist is required after operation \( add \).

v) Thus the refined privilege will be \( SA_{add} \rightarrow SA_{sign} \) for the medical record.

**Case 2 (dual roaming):** Assume \( DomainA.subject \) is a GP, who is asked to diagnose a patient at the patient’s home. The GP needs extra information about the patient and the information is stored in domain \( C \), represented by \( DomainC.object \). The GP has no direct access permit to the domain \( C \) and has to require the object via domain \( B \). The roaming processing steps are as follows.

- The GP sends a duty request to the subject server \( A \) which is
  \[
  D = \{ DomainA.subject, \text{read}, DomainC.object \}. 
  \]
- The subject server sees that the \( DomainC.object \) cannot be accessed directly and has to be accessed via domain \( B \). The subject privacy label sent from domain \( A \) to domain \( B \) is \( SPL = \{ SG = 9, SSG = 9, SSG.SA_{\text{read}} \} \). By applying the roaming rules specified in Figure 2.22, the roaming subject privacy label \( R(SPL) \) is set to
  \[
  \{ DomainA.subject.SG = 8, SSG = 8, SSG.SA_{\text{read}} \}. 
  \]
- As domain \( B \) has to obtain the patient’s record from domain \( C \), object roaming is required as well. The roaming object privacy label \( R(OPL) \) is
  \[
  \{ DomainC.object.OG = 6, OSG = 8, OSG.NP = \text{edit, distribution, manage}, OSG.SP = \text{comment, sign} \}. 
  \]

Then by adopting the above five step procedure from i) to v), the refined privileges are \( SA_{\text{comment}} \rightarrow SA_{sign} \).
2.5.3 Implementation

The diverse privilege controller of $PPDAC$ model was written in Java 1.6. Figure 2.23 shows the object server loading an XML file and processing it for the incoming request from the subject server. Binary code is used in the example, which represents subject activities read, edit, add, delete, comment, declare, replicate and manage; binary 1 indicates the represented activity is allowed while binary 0 indicates the represented activity is prohibited. Permission binary code 11001010 in this example allowing read, edit, comment and replicate but not other operations on the required target.

Figure 2.24 shows a simple subject server monitoring window. It shows the subject’s ID, required data and the subject’s basic control codes.

![Figure 2.23 The Object Server Demo](image.png)
Figure 2.24: The Subject Server Demo

Figure 2.24 demonstrates the subject server which sends a request for an object. It also shows the permission allowed for the second granular data which is called granular data 2. The text area displays the outputs of the subject server.

2.6 Discussion

2.6.1 Comparison with Existing Solutions

In Table 2.1, User diversity denotes the capability to support large number of unique subjects, roaming/reorganization is the capability of supporting subjects in multiple roles or changing position/job. Diverse privilege extensible refers to the mechanism of supporting personalized privileges for special environments. The term granular data levels indicates the capability of individual granular data access control: full support means flexibility to adjust granular data access levels as required, while
partial support means that either the granular data structure is fixed or cannot be adjusted to different application scenarios. The feature *dynamic data access levels* denotes the capability of supporting variable access levels in diverse data sharing environments: full support means that the data access level can be set as required and can be adjusted to a remote domain while partial support means only one, the former or the latter is supported. The term *cross-domain environments* denotes roaming-enabled features of the mechanism and the capability of identifying the responsible party when an access event happens in a cross-domain environment. Full support means an approach adjustable for diverse application scenarios including subject roaming and object roaming, and a clearly identifiable server responsible for enforcing data access restrictions. Partial support means any of object roaming, subject roaming or identification of access management responsibilities is supported, but not all of them.

*Table 2.1: Features Comparison*

<table>
<thead>
<tr>
<th></th>
<th>HASBE [42]</th>
<th>RMAMD [212]</th>
<th>Proposed Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. Subject privilege</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- User diversity</td>
<td>Full Support</td>
<td>Partial Support</td>
<td>Full Support</td>
</tr>
<tr>
<td>- Roaming/reorganization</td>
<td>Not Available</td>
<td>Partial Support</td>
<td>Full Support</td>
</tr>
<tr>
<td>- Diverse privilege extensible</td>
<td>Partial Support</td>
<td>Not Available</td>
<td>Full Support</td>
</tr>
<tr>
<td><strong>2. Granular data</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Granular data levels</td>
<td>Partial Support</td>
<td>Not Available</td>
<td>Full Support</td>
</tr>
</tbody>
</table>
HASBE [42] combines attribute-based solutions [202] and hierarchical access control [230] to support user hierarchy and unique users with diverse privileges. The model does not support access conditions and privilege relationships or constraints, such as ‘no privilege edit is granted for roaming users’. Also, for granular data management, the model in [42] lacks in support of access levels of granular data and conditions of granular data. Roaming is partially supported, which is benefited from the user hierarchy model, but the way the roaming process works was not clearly discussed in the paper. In addition, data roaming is not supported by the model [42].

RMAMD [212] proposed an enhanced roaming table mechanism. It built mapping links between roles in different domains. Once a domain in the path of a mapping link is not available, roaming is disabled. Roaming users are unique, lack support of conditions (also used as user attributes) and fixed roaming tables are not able to satisfy roaming needs. The following example illustrates the shortcomings of RHAMD. A marketing team leader in the home domain has privilege pr1, pr2 and pr3. When required to roam to an other department twice, for the first time the subject requires privilege pr1 and pr4 and for the second time he requires privilege pr1 and pr2, but all required privileges are denied due to a special condition, so only pr5 is granted. In this case, the RMAMD is not able to deal with the request.
CHAPTER 2 ACCESS CONTROL IN CROSS-DOMAIN ENVIRONMENTS

The limitation of the proposed model includes its requiring a properly managed level-based system as the model highly relies on appropriate administration of subject and object levels.

2.7 Summary

The focus of this chapter was on data access control using both diverse privileges and granular data in cross-domain environments. A mechanism for combined subject granular privilege control and object granular data access control was proposed, and the issues of cross-domain data sharing environments were also addressed.

The proposed method addressed the problems of access control of unique users with diverse privileges in cross-domain environments.

The proposed model enables large amount of unique users, which makes the model more practical for large organizations. The support of diverse privileges brings more flexibility of management to access control and a dual control mechanism, hierarchical user attributes and hierarchical object attributes, offers better access management for fine-grained granular data. In addition, the user activity sequence and multiple conditions gives more power of control over work flows.

In cross-domain scenarios, full support of dual roaming is embedded in the proposed model. It enables user roaming, data roaming and privilege adjustment. The model
maps roles in one domain into roles in another domain, and thereby avoids the need of an additional role assignment when a subject roams into another domain. A clear responsibility of access enforcement in cross-domain application can be identified, which helps when dispute occurs.

The main advantages of the proposed mechanism are in it supporting diverse user privileges and accommodating application conditions in both single-domain and cross-domain environments. The object server can control granular data access with cooperating subject servers. With the subject and object roaming mechanisms, both user and data re-deployment are properly handled. Furthermore, the proposed method clearly defines the responsibility for data management in a multiple domain environment.

The next chapter (chapter 3) turns the focus on the purpose of access when requests lodged.
Chapter 3

Purpose-based PPDAC

The previous chapter introduced a privacy preserving data access control model (PPDAC) that addressed the issue of unique users with diverse privileges in cross-domain environments. This chapter examines another perspective of privacy preserving access control. It deals with the purpose of accessing target data and how the requested data is intended to be used, and presents a solution to enhance the PPDAC model in terms of purpose translation and adjustability.

3.1 Introduction

To maintain proper data privacy, traditional access control methods that only focus on privilege management are not sufficient. Byun emphasized that privacy protection
CHAPTER 3 PURPOSE-BASED PPDAC

cannot be fully achieved by traditional access control mechanisms mainly for two reasons: i) traditional access control models focus on subjects’ privileges on objects, while privacy requirements are also concerned with the purpose an object is used for and ii) the comfort level of data usage varies from individual to individual [116]. Meanwhile, Yang et al. pointed out that a privacy requirement ensures that data can only be used for its intended purpose and an access purpose is compliant with the data’s intended purpose [117].

To enhance privilege control methods to consider purpose, a method called purpose-based access control was proposed [116, 122] and then widely adopted and extended [50, 97-99, 118-128, 130]. The original purpose-based access control (PBAC) built a bridge between role-based access control (RBAC) and subject intentions. Later works enhanced PBAC with conditional roles, obligations, usage, purpose hierarchy and purpose process [39, 99, 116-127]. However, purposes heavily depend on users, application domains and environments. The meaning of a subject’s purpose can relate to different user operations and may lead to distinct privileges in cross-domain environments, and ignoring such issues can cause privilege conflicts.

This chapter enhances the PPDAC model detailed in the previous chapter by adding a privilege-oriented purpose-based access control module. It includes an improved
CHAPTER 3 PURPOSE-BASED PPDAC

PPSPC component (section 2.4.1) that integrates PPDAC principles, a multiple attribute-based object control component and a privacy preserving privilege refinement component. The main contributions are that the enhanced PPDAC model enables an access control incorporating both subject and object control, fills the gap between purpose and privilege control in cross-domain environments, and thereby enables the use of purposes for granular data and purpose translation in cross-domain environments. Moreover, the chapter designs a hybrid access control approach which enables flexible control for large organizations.

3.1.1 Chapter outline

The rest of this chapter is organized as follows. Section 3.2 reviews existing privacy preserving methods that focus on purpose. Section 3.3 presents preliminary concepts on purpose-based theory that will be used in this chapter. Section 3.4 proposes a privilege-oriented purpose-based module for the PPDAC model. The verification of the proposed PPDAC model is presented in section 3.5. This is followed by section 3.6, which shows an example of the proposed module. Section 3.7 discusses the advantages of the proposed method and then the chapter is summarized in section 3.8.
3.2 Background

Role-based privilege control and purpose-based control are key concepts in access control assignment as they address two main aspects of privacy policies: who can access the target and with what privileges, and for what purpose. The former was discussed in the previous chapter, and the latter is examined in this chapter. In this section, existing approaches that consider purpose in access control systems are reviewed.

Purpose is one of the most essential components of privacy-preserving access control, and is a central concept in privacy protecting access control models [100, 125]. A formal purpose-based model (PBAC) was proposed in 2005 by Byun and Li [116]. In [116, 117], purposes are classified into two categories that are widely adopted in many other solutions as well [118-130]: intended purposes (IPs) and access purposes (APs). Intended purposes are purposes associated with data and express the data owners’ wish. Access purposes refer to the way the accessor wants to use a certain object. The PBAC model builds on these purpose principles. When a user submits a request, the access control system verifies whether the APs complies with the IPs of the requested data object: permits access if it does, otherwise denies the request. The key feature of PBAC is that it supports explicit prohibitions and organizes purposes in a hierarchical structure. Moreover, granular data object administration can be
achieved via associating IPs with a data object, which can be a whole table, a column in a table or a tuple in a table. Figure 3.1 shows an instance of a hierarchical purpose and Figure 3.2 shows an example of how PBAC extends RBAC [118].

![Hierarchy of Purposes](image)

*Figure 3.1: An Example of Hierarchical Purpose [118]*

![PBAC Roles Diagram](image)

*Figure 3.2: An Example of PBAC Roles [118]*

To improve the PBAC model by incorporating existing RBAC models, the authors of [116] extended their own work and presented an improved the PBAC model with
extra control on IPs [118]. It divided IPs into strong IPs (sIPs) that cannot be overridden and weak IPs (wIPs) that can be overridden, thereby giving more flexibility to purpose control. Similarly, the authors of [117] extended their previous work to improve management by a process flow control mechanism over IPs [122].

There is also a large volume of literature on improving PBAC in relation to purpose management. In [121], the authors added purpose extension [131] to PBAC in order to enhance purpose control. In [123], PBAC’s purposes are declared explicitly by the users themselves. The key feature of this approach is that the user purpose is determined in a dynamic manner, based on subject attributes, context attributes and authorization policies, but the solutions lacks support of object related attributes. In [124], the key feature is supporting prohibitions specifying that some data cannot be used for certain purposes. The approach in [125] is designed for a variety of purposes, including conditional purposes. The authors of [127] proposed a conditional role model based on PBAC, where users dynamically activate conditional roles that are associated with purposes. Another work [50] explores the connection between permissions and roles with respect to purposes. Also, the author points that out a subject should specifically assert the purpose of accessing data in a request. The method presented in [50] directly assigns purpose to subject roles and employs two
CHAPTER 3 PURPOSE-BASED PPDAC

components: server constraints determination and subject obligation determination. By using these components, conditions are also considered in the approach.

In addition to purpose management improvements, purpose translation is also discussed in the literature. In [119], the author proposes a personal information flow model that specifies a limited number of subject activities on each type of information. An alternative to intended purposes is proposed in [120]: the authors map each subject to a sequence of activities with personally identifiable information, in order to ensure such information is used solely for the intended purpose.

In summary, existing solutions explore purpose management and the connection between purpose and subject activities. To further look into purpose-based access control, two issues have to be addressed and yet have been overlooked. (i) Purposes are not only associated with subjects, but the object side also needs to be considered. In addition to the object owner’s wish (IP), data type and other data related attributes can affect IPs as well. (ii) Purpose heavily depends on users, application domains and environments. The meaning of a subject's purpose can be translated to other user operations and may lead to varied privileges in different environments.
3.3 Privilege-oriented Purpose-based PPDAC

This section first presents a privacy preserving data access control (PPDAC) model. After a general description, it focuses on a PPDAC functional component named privilege-oriented purpose-based module (Figure 2.4). Using this module, the complete PPDAC model is proposed. Also in this section, basic concepts and notation relating to this chapter are introduced and explained.

The model shown in Figure 2.4 depicts two main components that realize the proposed system, namely the subject server and the object server. Each component has two main function modules: granular privilege control and purpose-based control. In chapter 2, the granular privilege control modules were discussed; this chapter focuses on the purpose-based access control modules.

3.3.1 Basic Concepts and Notation

This section starts by introducing definitions and notation for the purpose-based module. For clarification, the essential concepts will also be explained where they are used in this chapter.
Definition 3.1 (Subject purpose): Subject purpose (SP) is the purpose associated with an activity sequence of a subject, and it indicates what the subject intends to use the object for after gaining access.

SP is composed of subject main purpose (SMP) and subject special purpose (SSP). SMP indicates the overall aim of the subject while SSP relates to certain granular data. For each SMP, there can be none, one or more SSPs, that is, SSP does not have to be provided when the subject lodges a request. SSP and SMP are related concepts. SSP, compared with SMP, refers to some detail of the requested granular object. An example is as follows. In a building some unknown chemical material is reported to be leaking. Investigators’ SMP can be “want to know chemical materials used in this building”. The requested documents contain information about the building, including chemical materials, building structure, management summaries, business related information, customer information and other, restricted information. By default, only the granular data about the chemical materials will be disclosed to the investigators, as that matches the SMP. An investigator may request access to another piece of granular data “building structure” for the purpose of “evacuation”. This is an SSP and is optional, as if there is no one in the building, this granular data will not be needed.
Definition 3.2 (Subject Obligation): Subject obligation (SO) is a subject role related constraint.

Subject obligation is assigned to a subject based on the subject role. An example of a subject obligation is “for marketing manager, the document marketing report must be signed if no more addition is required from marketing team”.

Definition 3.3 (Subject Server Constraint): Subject server constraint (SC) consists of subject server related access conditions, represented as a sequence of subject activities that must or must not be performed before or during an access.

Subject server constraints are assigned to a subject according to the subject server’s conditions. An example of SC is “access are denied for all incoming requests after 5 pm” (which may be due to scheduled maintenance or other special events).

Definition 3.4 (Object purpose): Object purpose (OP) represents the object owner’s wish regarding valid and invalid use of the object.

There are two mutually exclusive categories, allowed object purposes (AOP) and prohibited object purposes (POP). AOP denotes the only purposes that are allowed
for an object while \textit{POP} denotes the only purposes that are prohibited for an object. An example of \textit{AOP} is “\textit{read} permission for all employees in the company, \textit{edit} permission for employees in the marketing department”. An example of \textit{POP} is “no data roaming to other departments or other organizations”.

\textbf{Definition 3.5 (Object obligation):} An object obligation (\textit{OO}) is a sequence of subject activities on an object that must or must not be conducted before, during or after an access.

Object obligation is generated by the object server according to the object server access conditions. An example of \textit{OO} is “subjects who \textit{edit} the object must \textit{sign} a declaration form”.

\textbf{Definition 3.6 (Object Type):} Object type (\textit{OT}) describes an object’s usage called category, and related temporal and organizational constraints called object category specifications.

Category and specification are used for imposing object constraints. An example of \textit{OT} is “2012 Q1 Marketing report” where “report” is the object’s category, “2012”, “Q1” and “marketing” are specifications. An \textit{OT} is associated with one object, and
can contain more than one categories, and more than one specification can be assigned to each usage category.

An example constraint can be "files in sub-domain X" can only be accessed via sub-domain X. On one hand, object purpose (OP) denotes the owner's wish or requirement applying to those who want to access it, on the other hand, object obligation (OO) describes the object server's requirements. The object privacy label (OPL) used in section 2.4.2 is extended, by adding purposes, obligations and constraints.

### 3.4. Privilege-Oriented Purpose-Based Module

This module is designed to enable a purpose-based mechanism in the proposed PPDAC model. This section extends the functions of the three main components discussed in chapter 2, which are privacy preserving subject privilege control (PSPC), privacy preserving object control (PPOC) and privilege refinement (PR).

The module shown in Figure 3.3 consists of three functional components which are subject-based access control (SBAC), object-based access control (OBAC) and privilege refinement (PR). SBAC is extended from PSPC (chapter 2.4.1) and caters
for subject purposes, subject obligations and subject server constraints, OBAC is extended from PPOC (chapter 2.4.2) and provides support for object types, obligations and object purpose. PR determines the most appropriate privileges for users with regards to the given purpose. In Figure 3.3, SPL indicates the subject privacy label and OPL denotes the object privacy label, which will be introduced in sections 3.4.1 and 3.4.2, respectively.

![Privilege-Oriented Purpose-Based Module](image)

*Figure 3.3: Privilege-Oriented Purpose-Based Module*

The privilege-oriented purpose-based module (Figure 3.3) processes user requests. For each request, SBAC generates an SPL which gives all subject-based access control codes; while for the OBAC, OPL is generated that contains all access control codes regarding the object. Then, both SPL and OPL are forwarded to the PR component and permitted privileges are derived.
CHAPTER 3 PURPOSE-BASED PPDAC

3.4.1 Subject-based Access Control (SBAC)

SBAC has three layers (Figure 3.4): subject attributes assignment (SAA), subject privilege interpretation (SPI) and subject privacy label generation (SPLG).

The SAA layer derives subject role from the subject server’s database, extracts the subject’s purposes from user requests and obtains subject server constraints from the subject server (section 3.4.1.1). The SPI layer translates the attributes in SAA layer into subject activity sequence (SAS, definition 2.5) and adjusts the translations depending on different domains (section 3.4.1.2). The SPLG layer encapsulates all subject-based control codes into a subject privacy label (SPL, introduced in chapter 2).

The detailed contents and functions of the three layers are presented later in the sections below.
3.4.1.1 Subject Attributes Assignment (SAA) Layer

The SAA layer has three functions: subject role assignment (RA), subject purpose assignment (SPA) and subject constraint assignment (SCA).

RA is a functional unit dealing with Hierarchical PSPC processes. It receives subject role from the subject server’s user authentication database and obtains subject grade.
CHAPTER 3 PURPOSE-BASED PPDAC

(SG, definition 2.6), subject sub-grades (SSG, definition 2.8) and subject obligation (SO, definition 3.2) from an organization’s policy database. The SG and SSG will be sent directly to the SPLG layer, while the SO will be passed on to the SPI layer for further processing (Figure 3.4).

SPA extracts the subject purpose (SP, definition 3.1) hierarchy from the subject request, which contains subject main purpose (SMP, see explanation of definition 3.1) and subject special purpose (SSP, see explanation of definition 3.1). The SMP and SSP will be forwarded to the SPI layer for further processing.

SCA obtains the subject server constraints (SC, definition 3.3) from the subject server. The SC will be sent to the SPI layer together with SMP, SSP SO for translation.

The RA, SPA and SCA together to form a whole subject-based access control foundation. Each of them represents one factor that affects the access results. RA represents administrative control in an organization, such as the human resource department or the role management team; SPA represents the needs of subjects to perform their work and SCA represents constraints from the subject server. The designation of this three functional control units is provides subject-based three dimensional control, by considering the organizational environment that assigns roles,
CHAPTER 3 PURPOSE-BASED PPDAC

the access request originator’s aim (subject purpose) and server constraints (e.g. time restrictions).

3.4.1.2 Subject Privilege Interpretation (SPI) Layer

The subject privilege interpretation (SPI) layer translates subject purposes (SP), subject obligations (SO) and subject server constraints (SC) to subject activity sequences (SAS) (Figure 3.5). The translation process is described below.

A purpose hierarchy containing SMP and SSP is translated to an SAS representing the requested subject activities on the target object. The translation result depends on organizational access purpose policies and the results can vary due to varying policies in different organizations (domains). For example, the main purpose “prepare report for distribution” and the target object “customer data” can be translated to “read all fields of customer data” in one domain, while in another it may be translated to “read customer age, gender only”. Such variation can happen due to data content or policy diversity.
Subject obligation (SO) needs to be translated when the subject moves to another domain (roaming). This translation is performed on the subject server the subject sends the request to. When roaming (presented in chapter 2), all associated subject servers’ SO will be attached to the subject. For instance, when a technician in domain $D_A$ roams to domain $D_B$, $D_A$ gives subject obligation “add report” and $D_B$ gives subject obligation “roaming technicians can only read object granular data (OGD) titled maintenance manual”. Then both obligations will be translated to activity sequence “$SA_{add\ Oreport}$” and “$SA_{read\ OGD_{maint}}$.”
Subject constraint (SC) can take different forms, such as “no access request is accepted after 5pm” due to server maintenance or “no distribution” due to company policies regardless of the subject roles. These constraints will be translated to an SAS as well.

To instantiate the model, expression syntax can be used. To help understanding the model, let us use a simple syntax and explain it in an example. A request expression "subject S requests to do marketing through the file repository Repo" needs to be converted to formal language; such as "<subject S> check <file> existence {and} read <profile section> {with the purpose of} [marketing]" clearly specifies the subject’s activities. In this example,

- <> indicate the subject or object in a request,
- {} indicate built-in conditions or logical expressions, such as “and”, “or”, “with”,
- [] indicate purposes,
- italics denote activities

The main function of the SPI layer is to convert subject purposes, obligations and constraints into subject activity sequences that can be directly controlled by the model. Such translation can be managed via domain-based purpose-activity mapping tables,
CHAPTER 3 PURPOSE-BASED PPDAC

such as that shown in Table 3.1. Using “main purpose is marketing” as the example, the activity sequence is $S_iA_{check} \rightarrow [(S_iA_{read} \rightarrow S_iA_{deliver}) \leftrightarrow (S_iA_{finish} \rightarrow S_iA_{report})]$ {and} {time=10am to 3pm} which means for the subject $i$, the subject activity sequence is first check the condition whether the access time is between 10am and 3pm. If the condition is satisfied, then check availability of the requested object. If this object is available, then either perform “read the object and then deliver it” or “finish updating and then report to manager”.

In summary, the $SPI$ layer takes subject purposes, subject obligations and subject server constraints to build a connection between subject requests and subject activity sequences. It translates the subject’s request into a policy-manageable syntax. Moreover, it provides manageable privilege control elements for the next layer: the subject privacy label generation ($SPLG$) layer.

Table 3.1: Sample Activities Mapping Table

<table>
<thead>
<tr>
<th>Acts</th>
<th>Description of requests and purposes</th>
<th>Associated activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Preparing quarterly report for general manager</td>
<td>Read, copy, comment, sign</td>
</tr>
<tr>
<td>B</td>
<td>Prepare report for division manager</td>
<td>Read, edit</td>
</tr>
<tr>
<td>C</td>
<td>Marketing</td>
<td>List, read</td>
</tr>
<tr>
<td>D</td>
<td>Case study</td>
<td>List, read, log</td>
</tr>
</tbody>
</table>
CHAPTER 3 PURPOSE-BASED PPDAC

3.4.1.3 Subject Privacy Label Generation (SPLG) Layer

Subject privacy label generation (SPLG) is an extension of the subject privacy label (SPL) component in chapter 2, to encapsulate the subject grade, sub-grade, SP activity sequence, SO activity sequence and SC activity sequence into a single subject privacy label (SPL).

A typical SPL has the form $<domain.subject, \text{domain.subject.role, domain.subject.purpose, domain.subject.SG, domain.subject.SSG, domain.subject.SP.SAS, domain.subject.SO.SAS, domain.subject.SC.SAS}>$, where

- $domain.subject$ indicates the domain where the subject is located. This attribute affects subject obligations, subject grade, object purposes and object obligations.
- $domain.subject.role$ indicates the role assigned to the subject. This attribute affects subject grade, subject sub-grade, subject obligations and object obligations.
- $domain.subject.purpose$ is the purpose of the subject.
- $domain.subject.SG, SSG$ are access level attributes (subject grade and subject sub-grade) that are needed for privilege refinement and authorization.
- $domain.subject.SP.SAS$ denotes the subject activity sequence translated from the subject purpose.
CHAPTER 3 PURPOSE-BASED PPDAC

- *domain.subject.SO.SAS* denotes the subject activity sequence translated from the subject obligation.
- *domain.subject.SC.SAS* denotes the subject activity sequence translated from the subject server’s constraints.

3.4.2 Object-Based Access Control (OBAC)

The *OBAC* component implements object purpose-based access control. It supports object purpose (definition 3.4), object obligation (definition 3.5), object type (definition 3.6) and fine-grained object granularity (*OGD*, definition 2.7).

The *OBAC* consists of three processing layers: object attributes assignment (*OAA*) layer, object privilege interpretation (*OPI*) layer and object privacy label generation (*OPLG*) layer as shown in Figure 3.6.

The *OAA* layer obtains the object owner’s intension, stored in the form of object purpose (*OP*), object type (*OT*) and object obligation (*OO*), from the object server. These attributes will be then sent to the *OPI* layer and translated into activity sequences. These activity sequences will be passed on to the *OPLG* layer.
3.4.2.1 Object Attribute Assignment (OAA) Layer

The OAA layer caters for three attributes shown in Figure 3.6: object purpose (OP, definition 3.4), object type (OT, definition 3.6), and object obligation (OO, definition 3.5).
CHAPTER 3 PURPOSE-BASED PPDAC

OP is dealt with by the functional unit object purpose assignment (OPA). Object purpose represents the object owner’s intension regarding what the object can be used for. It is either allowed object purpose (AOP) or prohibited object purpose (POP). The purpose will be sent to the OPI layer for processing.

Object type assignment (OTA) deals with object types (OT) representing temporal and organizational constraints. OT is composed of category and specifications that are usually set by the object server or the owner. For example, an OT can be quarterly report, where report is the object category and quarterly is a specification qualifier of the report. The specification may be used to limit the object purpose and obligations based on privacy policies and management rules, such as a quarterly report can be updated within a week after it was submitted, and after such time the file will be automatically locked and will be set to read only. Also, it is possible to add extra specifications to the object, such as organizational attributes, e.g. marketing quarterly report, so that the key specification marketing affects object purpose and obligations, and in other departments only managers or above can access the marketing quarterly report.

The function unit object obligation assignment (OOA) deals with object obligations (OO) representing the constraints from the object server. These constraints are
different to those provided by \textit{OP} and \textit{OT}. The constraints brought by \textit{OO} may not apply to a specific object only. For example, an \textit{OO} ‘subjects who edit the object must sign a declaration form’ can be a requirement for all objects.

\textit{OPA}, \textit{OTA} and \textit{OOA} together form a holistic foundation of object-based access control, each of them represents one factor that affects the access of objects. \textit{OPA} expresses access limitation by the object owner; \textit{OTA} indicates inherent limitation of the data while \textit{OOA} focuses on constraints other than \textit{OPA} and \textit{OTA}. The access limitations will then be forwarded to the \textit{OPI} layer for translation.

Compared with \textit{PPOC} (section 2.4.2), the major feature of the \textit{OAA} layer is structuring object attributes into a hierarchy to facilitate the definition and administration of object purposes and obligations. For example, in an organization, managers handle data types and obligations, while data owners set object purposes.

\textit{3.4.2.2 Object Privilege Interpretation (OPI) Layer}

The object privilege interpretation (\textit{OPI}) layer translates object purposes (\textit{OP}), object type (\textit{OT}) and object obligation (\textit{OO}) to a subject activity sequence (\textit{SAS}) (Figure 3.7).
Object purpose contains either allowed (AOP) or prohibited purposes (POP) that will be translated to SAS representing the owner’s intension on what the object can be or cannot be used for.

Both OT elements, category and specifications, will be translated depending on the organization’s data classification policy that can change with time. For example, the old policy states “marketing report can be accessed by marketing department only” and after a new department being set up, the new policy can be “marketing report can be accessed by marketing department and customer service department only”. In this
example, although the OT remains the same, the translation has to be changed according to the updated policy.

The translation of OO is similar to that of subject constraints (SC). These three attributes OP, OT and OO will be translated into an activity sequence (SAS) in the OPI layer and then be passed on to the OPLG layer.

3.4.2.3 Object Privacy Label Generation (OPLG) Layer

The object privacy label generation (OPLG) layer encapsulates the output of the OPI layer into an object privacy label (OPL) and sends it to the privilege refinement component (section 3.4.3).

A typical OPL has the form <domain.object, domain.object.OG, domain.object.OSG, domain.object.OP.SAS, domain.object.OT.SAS, domain.object.OO.SAS>, where

- domain.object indicates the domain where the object is stored. This attribute affects object constrained privileges and OG, as each domain has its own policies that can constrain objects and pre-determined object grade.
- domain.object.OG and OSGx are access control attributes needed for privilege refinement that were discussed in chapter 2.
CHAPTER 3 PURPOSE-BASED PPDAC

- *domain.object.OP.SAS* denotes an activity sequence associated with either allowed or prohibited object purposes.

- *domain.object.OT.SAS* relates to object category and specification. This attribute affects the object grade (*OG*) and object sub-grade (*OSG*), as different data types can be assigned pre-determined object grades, object sub-grades (for granular data) and access conditions.

- *Domain.object.OO.SAS* denotes the translation of object obligations into an activity sequence.

### 3.4.3 Privilege Refinement (PR)

Privileges that allow subjects to perform activities are refined in four steps, as shown in Figure 3.8. Each step deals with one or more subject and object attributes and passes the results on to the next step.

Step (1) deals with the highest priority attributes: subject grade (*SG*), object grade (*OG*), subject main purpose (*SMP*) and allowed object purpose (*AOP*)/prohibited object purpose (*POP*). The evaluation of *SG* and *OG* was detailed in section 2.4.3. For *SMP*, the following rule applies.
**Rule 3.1:** Subject main purpose (SMP) is allowed only when SMP is an allowed object purpose (AOP) or SMP is not a prohibited purpose (POP).

If SMP satisfies **rule 3.1** and SG passes the algorithm detailed in section 2.4.3, the process moves to step (2). Otherwise, access is denied.

---

*Figure 3.8: PR Process Flow*
CHAPTER 3 PURPOSE-BASED PPDAC

Step (2) works on the granular data level and focuses on subject sub-grades, object sub-grades and subject special purpose (SSP). Evaluation of SSG and OSG has been detailed in section 2.4.3. It is not necessary to have SSP for each granular data items; if a subject does not specify any special purpose for granular data, rule 3.2 applies. If the SSP complies with the OP, and the result of SSG and OSG evaluation is a pass, step (3) will follow.

**Rule 3.2:** A subject special purpose (SSP) inherits its content from the subject main purpose (SMP) only when the subject does not specify SSP for the relevant granular data.

Step (3) works on combining subject constraints (SC), subject obligations (SO) and object obligations (OO) together. These three attributes aim at either limiting the subject privileges under certain conditions or requiring further activities. This step merges conditions and obligations within the same category. For example, the conditions "subject cannot access if out-of-office" and "cannot access between 6pm - 9am" are combined into "if time is between 9am - 6pm, and the subject is in-office, then allow further privilege processing, otherwise access denied."
CHAPTER 3 PURPOSE-BASED PPDAC

Step (4) focuses on privilege selection. The privilege refinement algorithm was detailed in section 2.4.3.

3.5 Model Verification

This section describes the formal model verification of PPDAC by using the verification tool Failure Divergence Refinement (FDR), build 2.83 for academic purposes.

FDR [132] is a model verification tool based on Communicating Sequential Processes (CSP) state machines, where CSP is a processing language used in describing process state switching. FDR has been widely used in formal model verification since 1996 [133], when Lowe found a man-in-the-middle attack in the Needham-Schroeder public key protocols [134]. The system’s correct operation is verified in this section.

The steps to set up the verification tool were as follows:

1. Build the abstract model based of the proposed PPDAC and process as specified in Figure 3.9, by using the CSP language.

2. Provide specifications for the FDR, which is by giving a valid requirement and the requester can retrieve the proper required information, otherwise return errors.
CHAPTER 3 PURPOSE-BASED PPDAC

3. Run the model check to see whether the proposed PPDAC model satisfies the specifications mentioned in step 2. If it does not, the model checker will provide a counterexample.

The whole process can be briefly described as follows: a subject submits its identity and request, and finally obtains the required object if the request is authorized. There are four participants in the process flow: subject, subject server and object server. The legend of Figure 3.9 is shown below.

<table>
<thead>
<tr>
<th>Legend</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>chk_sg</td>
<td>Evaluate SG and OG, SMP and OP</td>
</tr>
<tr>
<td>error</td>
<td>Error message, a common syntax in FDR indicates errors, authentication failure and any other unsuccessful process endings</td>
</tr>
<tr>
<td>get_obj</td>
<td>Retrieve object from the data server</td>
</tr>
<tr>
<td>get_pobj</td>
<td>Receive processed object</td>
</tr>
<tr>
<td>open_obj</td>
<td>Get the required granular data</td>
</tr>
<tr>
<td>process_obj</td>
<td>Remove unauthorized granular data and put in tracking seed if required by the security need</td>
</tr>
<tr>
<td>pl_mat</td>
<td>Evaluate SSG, OSG, access conditions and SAS</td>
</tr>
<tr>
<td>perform</td>
<td>Refine privilege on the requested granular data</td>
</tr>
<tr>
<td>req_get</td>
<td>Get object request</td>
</tr>
<tr>
<td>spl_gen</td>
<td>Subject privacy label generation</td>
</tr>
<tr>
<td>warning</td>
<td>A warning message indicates the failure of the SG and purpose validation results</td>
</tr>
</tbody>
</table>
Figure 3.9 FDR Processing Flow Chart
The object database only communicates with the object server, and hence, they are combined together in the modeling. The verification result is shown in Figure 3.10.

Figure 3.10 FDR2 Verification Result

Figure 3.10 indicates that the process flow defined by the PPDAC model has satisfied the requirement (in FDR, the requirement is called specification, see section 3.5 step 2) of the system and passed the sequence tests. To illustrate the information flow, the debug window is shown in Figure 3.11. The debug window at the right displays the data transferred in each step in the system during the verification test.
The left window in Figure 3.11 shows the system structure. By clicking each component node, the debug window gives the control codes that are processed within it. This helps to determine how control codes are processed and where errors occur.

### 3.6 Illustrating Example

To help understanding the *PPDAC* model, this section presents an example of access control in a cross-domain environment. It describes the involved parties (domains), participants (subjects and objects), code base (privilege tables and roaming rules) and a case that illustrates how the proposed model works.
CHAPTER 3 PURPOSE-BASED PPDAC

There are two organizations involved and shown in the Figure 3.12, each representing an individual domain with its own subject server and object server. Organization $A$ ($O_A$) has its own groups called sub-domains and denoted by $domainAA$ and $domainAB$. For example, organization $A$ ($O_A$) can be an international enterprise and organization $B$ ($O_B$) can be a customer and marketing analysis company. Figure 3.13 to Figure 3.17 described the organization policies that are used by the proposed model.

Figure 3.12: Illustrating Example

<table>
<thead>
<tr>
<th>Access grades</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Subject roles</strong></td>
</tr>
<tr>
<td>General Manager</td>
</tr>
<tr>
<td>Department Manager</td>
</tr>
<tr>
<td>Marketing staff</td>
</tr>
</tbody>
</table>

Figure 3.13: Access Grades
### Illustrating example legend

<table>
<thead>
<tr>
<th>Components in privacy labels</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DomainA.subjectX</td>
<td>subject X in domain A</td>
</tr>
<tr>
<td>DomainA.subjectX.SG</td>
<td>subject grade of subject X in domain A</td>
</tr>
<tr>
<td>DomainA.subjectX.constraints</td>
<td>subject X’s constraints for request an object via domain A</td>
</tr>
<tr>
<td>DomainA.subjectX.purpose</td>
<td>The subject’s purpose when requesting an object</td>
</tr>
<tr>
<td>DomainB.objectY</td>
<td>object Y in domain B</td>
</tr>
<tr>
<td>DomainB.objectY.OG</td>
<td>object grade of object Y in domain B</td>
</tr>
<tr>
<td>DomainB.objectY.OT</td>
<td>Object Y’s object type, including two attributes category and specification</td>
</tr>
<tr>
<td>DomainB.objectY.purpose</td>
<td>Is either AOP or POP</td>
</tr>
<tr>
<td>DomainB.objectY.obligation</td>
<td>States the constraint related to the object server</td>
</tr>
</tbody>
</table>

*Figure 3.14: Illustrating Example Legend*

### Activity list for the two organizations

<table>
<thead>
<tr>
<th>Activity</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>read</td>
<td>Read the specified granular data</td>
</tr>
<tr>
<td>add</td>
<td>Add information to the granular data. Note: this activity does not allow to remove information</td>
</tr>
<tr>
<td>remove</td>
<td>Delete information from granular data. Note: this activity does not allow to add information</td>
</tr>
<tr>
<td>edit</td>
<td>Add and remove information of granular data</td>
</tr>
<tr>
<td>comment</td>
<td>Comment to granular data. Note: this activity does not allow editing original granular data</td>
</tr>
<tr>
<td>sign</td>
<td>Sign an object or certain part of an object. This activity is used when someone approves or acknowledges something</td>
</tr>
<tr>
<td>distribute</td>
<td>Distribute to other domains’ subjects</td>
</tr>
<tr>
<td>manage</td>
<td>Manage the object</td>
</tr>
</tbody>
</table>

*Figure 3.15: Activity List For The Two Organizations*
### Object grades

<table>
<thead>
<tr>
<th>Object</th>
<th>Object grades and sub-grades</th>
</tr>
</thead>
<tbody>
<tr>
<td>report</td>
<td>OG=5</td>
</tr>
</tbody>
</table>

*Figure 3.16: Object Grades*

### Object Type Translation rules

<table>
<thead>
<tr>
<th>Attributes in OT</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>category=report</td>
<td>The “report” category means only the data owner can edit and have to be signed after every update.</td>
</tr>
<tr>
<td>Specification1=customer analysis</td>
<td>Only marketing department and customer service department are allowed to access</td>
</tr>
<tr>
<td>Specification2=2010 Q1</td>
<td>Once created, no edit permission after 2010 Q2.</td>
</tr>
</tbody>
</table>

*Figure 3.17: Object Type Translation Rules*

### Purpose translation rules

<table>
<thead>
<tr>
<th>Purposes</th>
<th>Activity in home domain</th>
<th>Activity for roaming subject</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketing</td>
<td>Read, add, remove, distribute</td>
<td>read</td>
</tr>
</tbody>
</table>

*Figure 3.18: Purpose Translation Rules*

**In the example** let us assume DomainA.subjectX is a marketing manager, who is requesting the latest customer analysis report *(ObjectY)* from domain B and wants to distribute this report to sub-domains within domain A.
CHAPTER 3 PURPOSE-BASED PPDAC

The marketing manager sends a duty request to the subject server in domain $A$ which is $D = \{\text{DomainA.subjectX, } SA_{\text{read}}, \ SA_{\text{distribute}}, \ SA_{\text{manage}}, \ DomainB.objectY\}$ together with purpose “marketing”. The subject server identifies the DomainA.subjectX as a marketing manager and sets DomainA.subjectX.SG = 6. The SPL is $\{\text{DomainA.subjectX, DomainA.subjectX. role = marketing manager, DomainA.subjectX.purpose=marketing, DomainA.subjectX.SG} = 6, \ DomainA.subjectX.SP.SAS = [(SA_{\text{read}} \rightarrow SA_{\text{distribute}}) \ \hat{\downarrow} SA_{\text{manage}}], \ DomainA.subjectX.SO.SAS=N/A, \ DomainA.subjectX.SC.SAS=N/A\}$

The object server sets DomainB.objectY.OG = 5. DomainB.objectY.OT contains the object category and specifications, which are represented as DomainB.objectY.OT.category = report, DomainB.objectY.OT.specification1 = customer analysis, DomainB.objectY.OT.specification2 = southeast suburbs and DomainB.objectY.OT.specification3 = 2013 Q1. Each component of OT has its privilege constraint for the subject who requests the object.

Then the OPL is $\{\text{DomainB.objectY, DomainB.objectY.OG} = 5, \ DomainB.objectY.POP. \ SAS = SA_{\text{edit}}, \ DomainB.objectY.OT.SAS = \{\text{if} <\text{subject=data owner}> \ \text{then} \ SA_{\text{edit}} <\text{objectY}> \ \text{and} \ SA_{\text{sign}} [\text{prepare report}], \ \text{if} <\text{domainA = marketing department} \ \text{or} \ \text{domainA = customer service department}> \ \text{then} \ SA_{\text{read}}$}
CHAPTER 3 PURPOSE-BASED PPDAC

[marketing {or} customer service], DomainB.objectY. OO.SAS = {if} <licence> {then}

$OA_distribute \{and\} OA_{manage}\}.

The privilege refinement process is:

i) Compare $SG$ and $OG$. Because $SG$ is greater than $OG$ ($6 > 5$), the subject is allowed to access the record’s general information such as the file identifier.

ii) Compare $SSG$ and $OSG$. Because the object does not have granular sections, no comparison of $SSG$ and $OSG$ takes place.

iii) Generate the constraints: there is no constraint from the subject side. $OT$ and $POP$ impose the following constraints, “{if} <subject=data owner> {then} OA_{edit}

$<objectY> \{and\} OA_{sign} \{prepare report\}; \{if\} <\text{domainA.subjectX} = \text{marketing department}\ {or\} \text{domainA.subjectX} = \text{customer service department}> \{then\} OA_{read}$

[marketing {or} customer service], $POP = OA_{edit}$”. In addition, the constraints from the object server object obligation indicate that distribution and management is allowed only if the $subjectX$ has valid license, represented as “{if} <license> {then} OA_{distribute \{and\} OA_{manage}\}”.

iv) If $subjectX$ has a valid license, the refined subject activity sequence ($SAS$) will be $(OA_{read} \rightarrow OA_{distribute}) \downarrow OA_{manage}$ for the report and the constraints from iii) apply; if the $subjectX$ does not have a valid license, the refined privilege will be $OA_{read}$.
CHAPTER 3 PURPOSE-BASED PPDAC

3.7 Discussion

The model proposed in this chapter enhanced the privacy preserving access control model of the previous chapter with purpose management; conditions and constraints can be imposed on what the accessed data will be used for.

The proposed model enables access control that considers subjects, objects, as well as purposes. The control module on the subject side is “three dimensional”, enabling access management from three perspectives, namely organizational positions and roles, access request with purposes and access restriction from the environment called subject constraints. Similarly on the object side, the model has a “three dimensional” object control module enabling object administration that considers the data owner’s intentions in the form of object purposes, manage organizational and data repository requirements that are expressed as object type and the environment by imposing object obligations. The model overcomes the limitations of existing solutions, such as lacking consideration of environment constraints and cooperation between subject side control and object control.

From the perspective of purpose, the proposed model considers both subject purpose and object purpose. Subject purpose can describe a user’s request along with access conditions given by the role, and called subject obligation, and by the environment,
CHAPTER 3 PURPOSE-BASED PPDAC

called subject constraints. Object purpose takes intended purposes from the object owner, and reduces the complexity of intended purpose control by permitting only the allowed purposes and prohibited purposes. This can dispense with professional IT knowledge of the data owners. In addition, the proposed model enables purpose translation in roaming scenarios.

The overall model converts purposes to conditional and sequential privileges, so that the various purposes can be controlled properly via a privilege control mechanism. The model reduces the possibility of conflicts between different translations when roaming occurs by decomposing a purpose into three parts that are easier to translate.

3.8 Summary

This chapter presents a purpose based access control model that incorporates a three dimensional subject and object control mechanism.

It supports hierarchical subject (user) purpose and object (data) purpose. For subject purpose, the proposed method supports user access purpose containing subject main purpose and subject special purpose, user obligations and server constraints; while for object purpose, it supports data type containing data category and specification, data
CHAPTER 3 PURPOSE-BASED PPDAC

owner’s purpose containing allowed purpose and prohibited purpose, and data server constraints. In addition, the model supports purpose adjustment in roaming scenarios.

It also support complex conditions along with privilege sequences in both subject control side and object control side.
Part II

Privacy Preserving for Published Data
Part I discussed data sharing with known recipients. This part of the thesis focuses on privacy preserving of data shared with unknown recipients, which is also called data publishing. For example, the Census Bureau publishes data regularly and healthcare organizations publish medical data for data modeling and research purposes.

This part investigates the problem of protecting privacy by modifying the data, while also maintaining data utility. Data privacy of published data is preserved if the adversaries are not able to derive the original data from the modified (e.g. perturbed) data or the re-constructed results are not close enough to the original data. Maintaining Data utility means preserving data distribution, data format and data range of the original data, so the modified data is still usable by unauthorized recipients. At the same time, an authorized data recipient should be able to restore the original data [48]. Chapter 4 reviews privacy protection of published data and related literature. It also introduces evaluation methods. Chapters 5 and 6 present two data privacy protection algorithms that are based on Chebyshev polynomials and fractal sequences, respectively. Attack resistance is examined in the Appendix.
Chapter 4

Data Privacy Protection for Data Publishing: Basic Concepts

This is a short chapter that introduces the concept of perturbation and explains how the methods proposed in the following two chapters are used.

4.1 Introduction

Protecting the privacy of individuals is a challenging task in today's world. The amount of individual information published by various data holders is continually increasing. Some organizations, such as governments and census bureaus, are required to make personal information available, while other organizations, such as hospitals, may want to
publish their data voluntarily for research purposes [154]. For example, a hospital may release its patients’ medical/healthcare records to data analysts to facilitate the building of a classification model. On the other hand, data publishers are prohibited by law from disseminating any person-specific information that compromises an individual's privacy. Therefore, a common precaution adopted by data publishers is to remove all explicit identifiers such as name, address and social security number to make the resulting data look completely anonymous [165].

Although explicit personal identifiers are usually removed before data is published, the rest of the data can still make the data owner identifiable. A study conducted by Sweeney [52] estimated that 87% of the population of the United States can be uniquely identified using the seemingly innocuous attributes of gender, date of birth and 5-digit zip code. Such identifiable attributes are termed as quasi-identifiers (QIs). Clearly, released data containing such information about individuals should not be considered anonymous. When such information is linked to a medical dataset that contains all the above information along with diagnosis and medication data, they together constitute sensitive information on individuals, which should not be leaked.

Accordingly, the data should be processed before being published so that it is resistant to privacy leakage while still offering maximum utility to data analysts by allowing various information to be derived from the processed data. A number of techniques have been proposed to maintain privacy. Traditional encryption methods, including homomorphic
encryption [146] prevent information leakage, but they compromise utility, because encrypted data usually cannot be used for data analysis. Alternatively, data publishers frequently apply data generalization techniques or data transformation algorithms for privacy protection.

The generalization technique works by substituting the original values of given attributes with more generalized ones, based on the generalization hierarchy built on top of each attribute's domain [136], such as a student in computer science department can be generalized to a student at RMIT University. However, most generalization techniques suffer from a significant drawback in that the processed data is not restorable [67, 166]. If authorized users require access to the data, their request has to be responded to without generalizing the data. To deal with such cases, data perturbation algorithms are employed, which allow the restoration of the original data. Data perturbation works by combining noise with the original data, by addition or by multiplication [171].

This part of the thesis focuses on the question of reducing the risk of privacy leak while maintaining data utility when data is made available to different users. It presents a solution that ensures that: (i) publicly available data preserves data privacy; (ii) analysts who are not authorized to access real data are still able to utilize the publicly available data; and (iii) authorized users have full access to the original data. This chapter proposes a data privacy-preserving framework based on data perturbation. The original data is modified by multiplication and addition in a way that preserves important features of the
data, which allows its use for general analysis. The method is fully reversible, so authorized users can restore the data to its original form without any information loss.

4.1.1 Chapter Outline

The rest of this chapter is organized as follows. Section 4.2 reviews the literature with regard to the relevant algorithms for data privacy, so as to ascertain their limitations and enable an attempt to overcome them. Section 4.3 introduces the proposed data privacy protection framework ($DP^2F$).

4.2 Background

This section first introduces the basic concepts and notation used in Part II. Then, it reviews the literature on three types of approaches to data privacy: generalization, anatomization and permutation, and perturbation.

4.2.1 Concepts and Notation

The data privacy usually relates to within a micro-table in which each row represents a subject such as a person, a project or a company and each column indicates a particular attribute of each subject, such as age, gender or personal income, or budget and bidder
for a project, turnover of a company etc. Two examples of micro-tables are shown in Table 4.1.

Table 4.1: Example of format for micro-tables

<table>
<thead>
<tr>
<th>pID</th>
<th>Age</th>
<th>Sex</th>
<th>Education</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>33</td>
<td>M</td>
<td>Bachelor</td>
<td>12K</td>
</tr>
<tr>
<td>2</td>
<td>29</td>
<td>F</td>
<td>Bachelor</td>
<td>12K</td>
</tr>
<tr>
<td>3</td>
<td>35</td>
<td>F</td>
<td>Master</td>
<td>20K</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>M</td>
<td>Master</td>
<td>18K</td>
</tr>
<tr>
<td>5</td>
<td>46</td>
<td>M</td>
<td>Bachelor</td>
<td>22K</td>
</tr>
<tr>
<td>6</td>
<td>37</td>
<td>M</td>
<td>Doctorate</td>
<td>25K</td>
</tr>
<tr>
<td>7</td>
<td>29</td>
<td>M</td>
<td>Bachelor</td>
<td>10K</td>
</tr>
<tr>
<td>8</td>
<td>32</td>
<td>F</td>
<td>Master</td>
<td>14K</td>
</tr>
<tr>
<td>9</td>
<td>40</td>
<td>F</td>
<td>Bachelor</td>
<td>18K</td>
</tr>
<tr>
<td>10</td>
<td>29</td>
<td>M</td>
<td>Bachelor</td>
<td>14K</td>
</tr>
<tr>
<td>11</td>
<td>34</td>
<td>F</td>
<td>Master</td>
<td>20K</td>
</tr>
<tr>
<td>12</td>
<td>40</td>
<td>M</td>
<td>Doctorate</td>
<td>25K</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>pID</th>
<th>Age</th>
<th>Education</th>
<th>Suburb</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[31-35]</td>
<td>Bachelor</td>
<td>3000</td>
</tr>
<tr>
<td>2</td>
<td>[26-30]</td>
<td>Bachelor</td>
<td>3041</td>
</tr>
<tr>
<td>3</td>
<td>[26-30]</td>
<td>Master</td>
<td>3000</td>
</tr>
<tr>
<td>4</td>
<td>[36-40]</td>
<td>Master</td>
<td>3065</td>
</tr>
<tr>
<td>5</td>
<td>[46-50]</td>
<td>Bachelor</td>
<td>3041</td>
</tr>
<tr>
<td>6</td>
<td>[36-40]</td>
<td>Doctorate</td>
<td>3065</td>
</tr>
<tr>
<td>7</td>
<td>[26-30]</td>
<td>Bachelor</td>
<td>3071</td>
</tr>
<tr>
<td>8</td>
<td>[31-35]</td>
<td>Master</td>
<td>3065</td>
</tr>
<tr>
<td>9</td>
<td>[36-40]</td>
<td>Doctorate</td>
<td>3071</td>
</tr>
</tbody>
</table>

In Table 4.1, attribute ‘Age’, ‘Sex’ and ‘Education’ can be used to identify individuals. These attributes are called quasi-identifiers (QI). Formally, a quasi-identifier is a set of attributes that, in combination, can be linked with external information to re-identify or reduce uncertainty about all or some of the subjects [165]. Strictly speaking, a successful data privacy attack results in the attacker being able to find additional information on an individual or reduce the uncertainty about individuals’ data from this attack. To reduce the chance of data privacy attack, many techniques focus on QI as this is one of the most important factors in launching privacy attacks [52].
One approach to fending off privacy attacks is data perturbation, in which the original data is combined with noise and its values are changed. This part of the thesis proposes perturbation methods, that keep the processed data readable and in the same form, e.g. if it is numeric before processing, it is still numeric afterwards.

### 4.2.2 Generalization

Each generalization operation hides some details in QI attributes by replacing some values with a parent value, and the replaced values are not disclosed. This section first looks at four generalization schemes for protecting published data. *Full-domain generalization* indicates that all values in an attribute are generalized to the same level, such as in *k*-anonymity [53, 66] and incognito [74]. For example, for the attribute *Career*, *Lawyer* and *Engineer* are generalized to *Professional*, and, *Career*, *Dancer* and *Writer* are generalized to *Artist*. In *k*-anonymity, if a subject has a particular attribute value, then at least (*k*-1) other subjects must have the same attribute value. While *k*-anonymity addresses a major issue, it also has some weaknesses. Machanavajjhala et al. presented two cases in which tables that satisfied *k*-anonymity did not protect privacy [67]. A homogeneity attack is possible if sensitive values lack diversity, and results in the sensitive values being revealed. To counter the attack, a stronger privacy scheme, called *l*-diversity, was proposed [68]. It requires every QI group having at least *l* different values for the sensitive attribute, and the proportion of each sensitive value in every QI group should be less than or equal to 1/*l*. While *l*-diversity is a stronger privacy scheme than *k*-anonymity, it still has limitations [166]. When the distribution of the sensitive data
in the overall data set is skewed, $l$-diversity, in fact, can increase the probability of identification [166]. Also, when the data in an $l$-diverse group are syntactically different but semantically similar, important information can be learnt by an adversary. The method $t$-closeness [166] addresses these issues by requiring that the difference between distribution within each group and that of the general data set should not be more than a threshold $t$. The second generalization category is called Sub-tree generalization. [75-78]. These solutions require only the same sub-tree elements to be generalized. For example, if Engineer is generalized to Professional, Lawyer will also be generalized to Professional, but Dancer and Writer can remain unchanged as they belong to the parent Artist, a different sub-tree. An improved sub-tree method has been proposed in [74], which keeps some siblings unchanged. For example, if Engineer is generalized to Professional, Lawyer can still remain unchanged and Professional refers to all jobs under this sub-tree except Lawyer. The third generalization scheme is called Cell generalization developed in [79, 80]. These solutions keep some values of an attribute unchanged while other values of the same attribute are generalized. For example, one Engineer is generalized to Professional and another Engineer can remain unchanged. The fourth generalization is called Multidimensional generalization [81-83] which considers multiple QI attributes as a tuple and each QI can be decided whether to be generalized independently. For example “engineer, male” can be generalized to “engineer, Any Gender” while “engineer, female” can be generalized to “professional, female”.
In summary, different generalization schemes bring distinct data utility and data distortion (data privacy). Most of them satisfy privacy requirements but provide insufficient data utility [136]. Generalization schemes are used as a baseline for comparison with the proposed methods, in the experiment as these schemes keep some original data features, such as data value range and data value form.

4.2.3 Anatomization and Permutation

Anatomization dissects the data and de-associates the relationship between QI and sensitive attributes rather than modifies QI. The approach in [87] divides the original data into two separate tables: QI table containing QI attributes and sensitive data table containing sensitive data attributes. Both QI table and sensitive data table have only one common attribute called group ID. The values in the same group will be linked then the data can be used.

Permutation proposed in [88, 151] de-associates the relationship between a QI and a numerical sensitive attribute by partitioning a set of data records into groups and shuffling their sensitive values within each group. Another form of anatomy is called data table fragmentation, which divides original data table into several tables and links them via certain chosen attributes [143, 144].

As anatomization and permutation do not modify the original data, they may still need support from a generalization scheme such as $k$-anonymity and $l$-diversity. Also, without
both tables being available at the same time, data utility reduces significantly; while with both tables released together, privacy protection cannot be maintained without an other privacy protection method.

4.2.4 Perturbation

There are many data perturbation methods. Data re-ordering [150] and nearest neighbor data-substitution [167] techniques work by substituting values of the same attribute, while approaches in [79, 87] divide the original table into several sub-tables and re-group the sub-tables [79, 87]. Rotation-based transformation is usually applied in multi-dimensional space and transforms the whole data set [147, 174], or different sub-tables by using different parameters [148, 168, 175], to another form while still keeping the Euclidean distance between each pair of values. As the names suggest, additive perturbation adds noise to the original data [89-92] while multiplicative perturbation multiplies the data by some noise to hide sensitive information [149, 152].

The re-ordering, re-grouping and data splitting techniques are vulnerable to data linkage attack as they did not modify original data values. Rotation-based techniques lose data utility by changing data format, data value range or distribution. Additive and multiplicative perturbation techniques are either vulnerable to data reconstruction attack or lose data utility [160, 161].
This thesis proposes two methods for generating noise that can be combined with the original data for perturbation, and be removed afterwords in the restoration phase. The proposed methods use a hybrid perturbation mechanism to overcome the drawbacks in existing solutions.

4.3 Data Privacy Protection Framework

In the following a privacy protection framework is described, that can employ the perturbation methods proposed in chapters 5 and 6. To implement the proposed framework, the data is assumed to be numerical and stored in micro-tables. The perturbation algorithms work on one attribute stored as a column in the micro-table and is represented as a vector $A = [a_1, a_2, ..., a_M]^T$, where $a_i \in [p, q], i \in [1, M]$, $M$ is the number of records/individuals (rows), and $p$ and $q$ are the bounds of the attribute. Typically, the original data has a pre-determined format and value range, such as age should be from 1 to 99, disease code should be within a certain range etc.

The proposed framework is shown in Figures 4.1 and 4.2. Figure 4.1 shows the perturbation process flow and Figure 4.2 depicts the process in which the original data is restored from the perturbed data. The heart of the proposed method is the data perturbation algorithm that takes the original data and transforms it to similar data in the same format. The perturbation parameters are the key used for both the data perturbation and restoration, in the same way as in symmetric-key encryption. In order to keep the
difference between the original and perturbed values within a well-defined range, data scaling is used.

In the first step, the perturbation noise is generated and a privacy-preserving transformation is applied. In the proposed framework, two individual perturbation noise values are calculated for each data item in the series, and then combined with the original data; one noise is for multiplicative perturbation and the other for additive perturbation. The second step is scaling the perturbed data to ensure that the perturbed and original data will be in the same value range. The reason to use a hybrid method is that compared to either additive or multiplicative methods, hybrid ones have better resistance to data reconstruction attack methods [162].

To restore the original data, the perturbation noise is recalculated, the scaling is reversed and the privacy-preserving transformation is inverted. As both scaling and the privacy-preserving transformation are lossless operations, the original data can be accurately restored.
Figure 4.1: Proposed DP2F Perturbation Process Flow

Figure 4.2: Proposed DP2F Data Restoration Flow
Chapter 5

Chebyshev Data Perturbation

5.1 Introduction

This chapter presents a data privacy preserving method called Chebyshev data perturbation (CDP). CDP implements the data perturbation part of the Data Privacy Preserving Framework ($D^2P$), presented in chapter 4. Hybrid data processing is used, which comprises an additive part and a multiplicative part. Both parts use Chebyshev polynomials to generate initial noise sequences. Such sequences are then scaled, based on the perturbation parameters and the features of the original dataset, to ensure that the perturbation values fall into the same value range as the original data values.
The proposed method is evaluated in terms of data utility and information added by the perturbation noise. Attack resistance tests are also presented in the Appendix.

The distinguishing features of the proposed method are the following. i) It is able to keep the perturbed data in the same value range as the original data, and so it is computationally hard to distinguish the original from perturbed data; ii) the data utility in terms of data distribution is maintained; iii) it resists two classic data privacy attacks.

The rest of this chapter is organized as follows. Section 5.2 introduces the mathematical fundamentals. In section 5.3, the proposed method will be detailed via perturbation values, scaling perturbation values, the perturbation process and the restoration process. Section 5.4 implements the evaluation methods introduced in chapter 4 and evaluates the proposed method by these methods in terms of added information and data utility. Section 5.5 discusses the proposed method against existing solutions and section 5.6 summarizes the chapter. Attack resistance experiments are described in the Appendix.
5.2 *Mathematical Foundations—Chebyshev Polynomials*

To assist the generation of perturbation sequences, Chebyshev polynomials of the first kind are adopted. These are first introduced in this section, followed by a discussion of the reasons why they are used in the proposed method.

Chebyshev polynomials of the first kind are defined by the recurrence relation shown in equations (5-1).

\[
\begin{align*}
T_0(x) &= 1 \\
T_1(x) &= x \\
T_{n+1}(x) &= 2xT_n(x) - T_{n-1}(x) 
\end{align*}
\]  

(5-1)

Each polynomial degree leads to a differently shaped curve. Figure 5.1 illustrates the polynomial curve for some degrees of \( n \) in the \([-1, 1]\) interval.
Chebyshev polynomials of the first kind are used in the perturbation algorithm for the following reasons.

i) within the range of -1<x<1, the value of \( T_n(x) \) is in the range [-1, 1], and

ii) for a given value of \( T_n(x) \), the original \( x \) cannot be calculated without knowing the polynomial degree \( n \) [179];

In the rest of this chapter, the term Chebyshev polynomials is used instead of the full title of Chebyshev polynomials of the first kind.
5.3 Proposed Method - Chebyshev Data Perturbation (CDP)

This section first provides an overview of the Chebyshev Data Perturbation (CDP) process flow, components, assumptions and scenarios. Then it details the proposed hybrid perturbation algorithm. Hybrid perturbation indicates the combination of additive and multiplicative techniques, that is, perturbation values are added to and multiplied by the original data. This section concludes with the restoration process.

5.3.1 Overall Process Flow

Figure 5.2 shows the CDP data perturbation process which is explained as follows. First, two Chebyshev polynomials are generated, with the polynomial degrees of \( n_1 \) and \( n_2 \); the generated polynomials are represented by \( N_1 \) and \( N_2 \). Then, \( N_1 \) and \( N_2 \), together with four perturbation parameters (\( \alpha, \beta, \gamma \) and \( \delta \)) and segmentation parameter \( k \), are used to calculate perturbation values. Finally, the perturbation values are merged with the original data.

To restore the original data, the same perturbation noise is calculated again, and is removed from the perturbed data.
5.3.2 The Proposed Chebyshev Perturbation Algorithm

To facilitate the mathematical treatment of the proposed perturbation algorithm, it is assumed that the data to be privacy protected is a series of items, such as a row or column in a micro table. Treating this data as a vector, the calculations are performed on this vector. The basic perturbation equation takes the form of

\[ PD = OD \times PN_1 + PN_2, \]

where \( PD \) is the perturbed data, \( OD \) is the original data, \( PN_1 \) and \( PN_2 \) are perturbation noise calculated as follows. \( PN_1 = [(SF_i \times N_i) + (1 - SF_i)] \)
CHAPTER 5 CHEBYSHEV DATA PERTURBATION

and \( P N_2 = SF_2 \times N_2 \), where \( N_1 \) and \( N_2 \) are calculated using Chebyshev polynomials of the first kind, but the degree of the polynomials is different for the two components; while \( SF_1 \) and \( SF_2 \) are scaling factors to keep the perturbed data values in a defined range. The main advantage of the Chebyshev polynomials here is that their values oscillate between +1 and -1 on the \([-1, +1]\) interval. Note: \( PN_i \) must never be zero to ensure reversibility of the process. If the calculations produce a zero for \( PN_i \), it is replaced by a preliminarily agreed value that is used for multiplication during perturbation and for division in the restoring phase.

5.3.2.1 Calculating the Perturbation Values

The actual values of the perturbation noise are used for both data transformation and restoration. The calculation of these values is performed in three steps. This section first introduces the overall calculation process and then explains each step in detail.

The overall calculation process can be described as follows. First, the original data series is divided into \( k \) groups or subvectors, as shown in Figure 5.3. The number of data elements need not be the same in each group, for example, group 1 may have 5 data elements, group 2 may have only 4 data elements, group 3 may have 5 data elements again, and so on. Then each group is linearly mapped to the \((-1,+1)\) interval,
i.e. each data element is mapped to a number between -1 and +1, and the Chebyshev polynomial’s value is calculated at each mapped point as shown in Figure 5.3. This polynomial value is used to calculate the perturbation noise. Three parameters \((\alpha, \beta, \gamma)\) are introduced for information hiding and one \((\delta)\) for scaling. The first parameter, \(\alpha\) is used to shift the Chebyshev polynomial along the x axis, the second parameter, \(\beta\) is used to compress the polynomial along the x-axis, and the third parameter, \(\gamma\) is used to compress (or expand) the polynomial along the y-axis. Another parameter \(\delta\) is used for scaling, so that the perturbed data remains in the same value range as the original data.

The calculation of the perturbation values involves three steps:

i) division of the original data into groups;

ii) calculation of the perturbation values for each original data item (vector element), and

iii) performing the perturbation and scaling operations. Here, a data item is an attribute record presented as \(a_i\). Each of these steps is explained below.
Figure 5.3: Process for Obtaining the Perturbation Vector
CHAPTER 5 CHEBYSHEV DATA PERTURBATION

Step 1 Data grouping

The original data is divided into $k$ groups or subvectors as follows. Division length ($DL$) is defined as $DL = M/k$, where $M$ denotes the number of the original data elements and $2 \leq k \leq \frac{M}{2}$. Let group $i$ have $t_i$ number of elements (see Figure 5.3), and $t_0 = 0$. Then $t_i = floor(DL)$, where $floor()$ represents the integer part of a number, $t_2 = floor(2 \times DL) - t_1$ and $t_i = floor(i \times DL) - \sum_{j=1}^{i-1} floor(t_j \times DL)$, where $1 \leq i \leq k$. The number of elements in the different groups may not be the same, but this has no effect on the perturbation calculations. In this way, the original attribute vector $A_r$ is divided into $k$ number of subvectors, which can be expressed as $A_r(i) = \{a_{(i,j)} | 1 \leq i \leq k, 1 \leq j \leq t_i\}$, where $a_{(i,j)}$ denotes the $j$-th element in group $i$ of the original dataset. Parameter $\beta$, where $0 < \beta < 1$, is a compression factor that is used to map the range $[-1,+1]$ to $[-\beta, +\beta]$. In the whole process, element $j$ in group $i$ is associated first with a point between -1 and +1, and subsequently with a point between $-\beta$ and $+\beta$. The point in the $[-\beta, +\beta]$ interval which corresponds to $a_{i(j)}$ is denoted as $x_{ij}$ and is given by the formula $x_{ij} = -\beta + \frac{2\beta j}{t_i}$.

Step 2 Perturbation noise calculation:

The perturbation noise is calculated for each element $a_{i(j)}$ in each group. Group $i$ has $t_i$ elements, and for each element $a_{i(j)}$ in interval $i$, two perturbation values
cdp1\(_i(j)\) and cdp2\(_i(j)\) are calculated as shown in equations (5-2) and (5-3). The first value, cdp1\(_i(j)\) is for the multiplicative component and cdp2\(_i(j)\) is for the additive component of the perturbation. In the calculations, the polynomial values are calculated at a modified \(x'_{ij}\) point, which is calculated as \(x'_{ij} = -\beta + \frac{2\beta j}{t_i + \alpha}\). The factor \(\alpha\) is introduced as an additional security parameter that is known by authorized users only. The effect is a shift of the \(x\) values in the negative direction, and the magnitude of the shift changing from zero at \(x = -\beta\) to \(\frac{\alpha}{t_i + \alpha}\) at \(x = +\beta\) in a linearly decreasing fashion. By decreasing the shift magnitude in this way, I can keep the shifted value in the \([-\beta, +\beta]\) interval. The actual perturbation values are calculated by the following formulas: cdp1\(_i(j)\) = \(T_n(x'_{ij})\) and cdp2\(_i(j)\) = \(T_{n+1}(x'_{ij})\). Here, \(T_n\) is the Chebyshev polynomial of degree \(n\), while \(j\) indicates the index of an element within group \(i\). To reduce possible correlation between cdp1 and cdp2, the degrees of the Chebyshev polynomials were chosen so that one polynomial is not a divisor of the other; \(T_n\) and \(T_{n+1}\) satisfy this criterion according to [180].

Inserting the values of \(x'\) in the formulas equations (5-2) and (5-3) are obtained.

\[
\text{cdp1}_i(j) = T_n(-\beta + \frac{2\beta j}{t_i + \alpha}) \quad 1 \leq j \leq t_i \tag{5-2}
\]

\[
\text{cdp2}_i(j) = T_{n+1}(-\beta + \frac{2\beta j}{t_i + \alpha}) \quad 1 \leq j \leq t_i \tag{5-3}
\]

The values of cdp1\(_i(j)\) and cdp2\(_i(j)\) are in the range of \([-1, +1]\), as \(0 \leq \beta \leq 1\).
CHAPTER 5 CHEBYSHEV DATA PERTURBATION

Step 3: Scaling the noise

The utility of the perturbed data can be improved if some statistical parameters of the original data are maintained in the perturbation process. This requires additional processing that restores these characteristics of the data. Two cases are presented here. First, the perturbed data has the same mean as the original; in the second case, maximum difference between the perturbed and original data is kept within well-defined limits. In both cases, two scaling factors $\gamma$ and $\delta$ are used to achieve the aim. The next section explains how the original data is perturbed.

5.3.2.2 Perturbation

This section presents two perturbation methods. One is to maintain the mean of the original data and the other is to limit the difference between the perturbed and original data.

To maintain the mean of the original data, two scaling factors $\gamma$ and $\delta$ are introduced as follows.

$$\gamma = \frac{\sum_{i=1}^{k} \sum_{j=1}^{l} a_i(j)}{\sum_{i=1}^{k} \sum_{j=1}^{l} [a_i(j) \times (1 - cdp l_i(j))] } \quad (5-4)$$

$$\delta = \frac{\sum_{i=1}^{k} \sum_{j=1}^{l} |a_i(j) - \mu_i|}{\sum_{i=1}^{k} \sum_{j=1}^{l} cdp l_i(j)}$$
The multiplicative and additive parts can be calculated using formulas (5-6) and
(5-7).

\[ pt1(j) = cdp1(j) \cdot \gamma + (1 - \gamma) \]  
(5-6)

\[ pt2(j) = cdp2(j) \cdot \delta \]  
(5-7)

The perturbed data are then calculated by equation (5-8).

\[ a_i'(j) = a_i(j) \cdot pt1(j) + pt2(j) \]  
(5-8)

Based on equations (5-4) to (5-8), the mean of the perturbed data is

\[ \frac{\sum_{i=1}^{k} \sum_{j=1}^{l_j} a_i'(j)}{M} = \frac{\sum_{i=1}^{k} \sum_{j=1}^{l_j} a_i(j) \times [cdp1(j) \cdot \gamma + (1 - \gamma)] + \sum_{i=1}^{k} \sum_{j=1}^{l_j} cdp2(j) \cdot \delta}{M} \]

\[ = \frac{-\sum_{i=1}^{k} \sum_{j=1}^{l_j} a_i(j) + \sum_{i=1}^{k} \sum_{j=1}^{l_j} a_i(j) + \sum_{i=1}^{k} \sum_{j=1}^{l_j} a_i(j)}{M} = \frac{\sum_{i=1}^{k} \sum_{j=1}^{l_j} a_i(j)}{M} \]

Which is the same as that of the original data.
The second method limits the difference between the perturbed and the original data. In this case, the values of scaling factors $\gamma$ and $\delta$ can be directly assigned, and the additive and multiplicative parts can be calculated using the same formulas as the first method, equations (5-6) and (5-7). In the second method $\gamma$ and $\delta$ are the perturbation scale controllers; they scale the $cdp,(j)$ in order to limit the magnitude of the perturbation noise. The perturbed data are then calculated by equation (5-8).

5.3.2.3 Restoration

In order to accurately restore the original data, all parameters have to be correctly procured, i.e. the Chebyshev polynomial degrees $(n_1$ and $n_2$), data segmentation parameter $(k)$ and scaling parameters $(\alpha, \beta, \gamma$ and $\delta$) have to be known. The restoration steps are listed below and are shown in Figure 5.4.

1. Initialize factors and parameters, and derive the perturbation Chebyshev polynomials.

2. Calculate division intervals and compute both additive and multiplicative parts according to Section 5.3.2

3. The output sequence from step 2 is applied on the perturbed data to restore the original data, based on equation (5-9).
\[ a_i(j) = \frac{a_i'(j) - pt2_i(j)}{pt1_i(j)} \]  

\[ 5 \text{ - 9} \]

**Figure 5.4: Restoration Process**

### 5.4 Experiments and Results

#### 5.4.1 Evaluation and Experimental Setup

This subsection describes how the evaluation methods, introduced in chapter 4, are used to examine the proposed method and the setup of the experimental environment.
The datasets in the experiments followed particular distributions and were generated randomly.

5.4.1.1 Distribution Tests

The number of data samples was $M=4000$, and their value was in the range of 30 to 50 for the normal distribution and 25 to 65 for the uniform distribution. For the proposed method, three sets of the parameters were chosen as follows.

i) $k=11, n=13, \alpha=4.7, \beta=(n+5)/(n+6), \delta=2, \gamma=0.05$, results in $PA \approx 5\%$.

ii) $k=11, n=13, \alpha=4.7, \beta=(n+5)/(n+6), \delta=5, \gamma=0.1$, results in $PA \approx 10\%$.

iii) $k=11, n=13, \alpha=4.7, \beta=(n+5)/(n+6), \delta=10, \gamma=0.5$, results in $PA \approx 20\%$.

For the comparison, the generalization technique used 5 as value intervals, such as [25-29], [30-34], etc. The parameters for the proposed method were chosen as reasonable values for the generated dataset.

The experiments were carried out on an Intel® i7-3770 machine with 16G RAM on a Linux Fedora operating system, and Matlab was used for data generation in all tests.
CHAPTER 5 CHEBYSHEV DATA PERTURBATION

5.4.1.2 Empirical Information Content Tests

Information content is a common metric of a dataset [154, 184], and is calculated according to equation (5-10),

\[ I(X) = -\log_2 P(x_i) \tag{5-10} \]

where vector \( X \) has \( x_1, x_2, ..., x_M \) total \( M \) elements (original data samples) and \( P(x_i) \) denotes the probability to have \( x_i \) identified [184]. By measuring information content added to the data, in other words the distortion of the data, we can characterize the effectiveness of a particular data protection method.

Assuming each data item has the same probability to be identified, the information content for the whole data set can be expressed as equation (5-11).

\[ W(X) = -M \log_2 P(x_i) \tag{5-11} \]

For the original data, as no added information is involved, \( W(X) = 0 \). For \( k \)-anonymized data set, the added information content of the data set is \( M \cdot \log_2 k \).

For \( l \)-diversity on \( k \)-anonymized data set, the information content is calculated by \( M \cdot \log_2 k \cdot l \) and in the tests, \( l = k \).
For the proposed method, each data is perturbed by the hybrid perturbation method and the noise value added is $pt_i(j)$ denoting the perturbation value for the $j$-th element in the $i$-th interval. Let $Max(pt)$ denote the maximum difference between the original data and the perturbed data value. Given a perturbed data $a'_i(j)$, the original data is $a_i(j) \in [a'_i(j) - Max(pt_i(j)), a'_i(j) + Max(pt_i(j))]$. Therefore, $P(x_i)$ can be calculated as $\frac{1}{2Max(pt_i(j))}$. Then, the information content for the proposed method can be calculated as $M \cdot \log_2 2[Max(pt_i(j))]$.

The information content tests were carried out three times and each of them uses the same parameters as those in the distribution tests. These experiments not only compare the results of different methods, but also show the impact of the proposed method’s parameters on information content. The parameter $k$ in $k$-anonymized data set is 5 and $l = k = 5$ for $l$-diversity.

5.4.2 Experimental Performance

This section shows the experimental results generated by the evaluation methods presented in section 5.4.1. Two experiments are described in this section, which are the distribution and information content. Attack resistance is described in the Appendix.
Figures 5.5 to 5.10 depict the distribution of the original, perturbed and generalized data. Three sets of parameters were used and each set generated one normal distribution and one uniform distribution.

5.4.2.1 Distribution Tests

Figure 5.5: Distribution Test -- Normal Distribution with PA ≈ 5%
Figure 5.6: Distribution Test -- Uniform Distribution with PA ≈ 5%

Figure 5.7: Distribution Test -- Normal Distribution with PA ≈ 10%
Figure 5.8: Distribution Test -- Uniform Distribution with PA ≈ 10%

Figure 5.9: Distribution Test -- Normal Distribution with PA ≈ 20%
As expected, Figures 5.5 and 5.6 show the distribution of the perturbed data is almost the same as that of the original data, when the PA is approximate 5%. From the Figures 5.7 and 5.8, it can be seen that the distributions of the perturbed data, obtained by using the proposed method, and the original data still closely follow each other, while when the PA reached approximate 20%, the distribution of the perturbed method deviated from the original data. In all distribution tests, the generalization method changes the distribution.

Figure 5.10: Distribution Test -- Uniform Distribution with PA ≈ 20%
CHAPTER 5 CHEBYSHEV DATA PERTURBATION

5.4.2.2 Information Content

The information content tests show the added distortion to the original data by the proposed method, $k$-anonymity and $l$-diversity. In the tests, different data sizes were chosen so that the trend also can be seen from the diagram. The parameters used for the proposed method in these tests were the same as that of the distribution tests.

Figures 5.11 to 5.13 show that the proposed algorithm significantly increases the distortion and outperforms $k$-anonymity or $l$-diversity when the magnitude of the perturbation noise is at least 10\% of the original data. Even when the noise is only 5\%, the proposed method still outperforms $k$-anonymity.

Figure 5.11: Information Content Test with $PA \approx 5\%$
Figure 5.12: Information Content Test with PA ≈ 10%

Figure 5.13: Information Content Test with PA ≈ 20%
5.4.2.3 Attack Resistance

Attack resistance is examined in the Appendix.

5.5 Discussion

The proposed perturbation method provides a significant contribution in the data privacy preserving area by being able to maintain data utility in terms of (i) the perturbed data closely follows distribution as that of original data, (ii) the data format is kept and (iii) the data value range is kept. And above all, it provides data privacy. Compared to the generalization technique, although both of them are able to protect data privacy, the generalization method impairs data utility in terms of data distribution while the proposed method is able to maintain it. In addition, other methods are vulnerable to classic data reconstruction (SPF and BE-DR) attacks [145, 162]. On the other hand, the proposed method not only maintains data utility but also resists these two attacks.

As Figures 5.5 to 5.8 illustrate, the transformed values maintain almost the same distribution as the original data. Figures 5.11 to 5.13 show the information content added by the processed data of different methods. The experiments showed that while \( PA \) is equal to 5\%, the proposed method is not as good as \( l \)-diversity and the actual
noise added by the perturbed value is very small. When $PA$ reaches 10%, the proposed method has better results than both $k$-anonymity and $l$-diversity. Under the same circumstance, the proposed method can still maintain the data distribution, as shown in distribution test. The proposed method can also keep the perturbed data in the original value range, and so the perturbed data cannot be distinguished from the original data.

A comparison of the various features of the proposed method with other techniques in the literature is summarized in Table 5.1.

<table>
<thead>
<tr>
<th></th>
<th>Privacy protection method</th>
<th>Proposed CDP method</th>
<th>Existing perturbation methods</th>
<th>Generalization methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perturbed data range</td>
<td>Can be fixed within a range</td>
<td>Arbitrary</td>
<td>Can be fixed within a range</td>
<td></td>
</tr>
<tr>
<td>Privacy protection</td>
<td>High level</td>
<td>Medium level</td>
<td>Medium level</td>
<td></td>
</tr>
<tr>
<td>Flexibility for adaptation</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Robustness to attacks</td>
<td>High</td>
<td>Low</td>
<td>Medium</td>
<td></td>
</tr>
<tr>
<td>Data utility of perturbed data</td>
<td>Medium – High</td>
<td>Medium</td>
<td>Low to Medium</td>
<td></td>
</tr>
</tbody>
</table>
5.6 Summary

In this chapter, a hybrid, multiplicative and additive data perturbation method was proposed to protect the privacy of published data. The perturbation maintains data utility by keeping certain characteristics of the original data. This chapter presented two options: the first maintained the mean of the original data, while the second kept the amplitude of the perturbation (the ratio of the original to perturbed data) within limits determined by the user. As the perturbation is reversible, authorized users who know the perturbation parameters can restore the original data. Unauthorized users who do not know the parameters cannot restore the original values, but still utilize the perturbed data.

The quality of perturbation is measured as the added distortion. It was shown that the method performs better than others and considerably increases the distortion when the perturbation noise is large. In the case of smaller perturbation noise, the increase is smaller while the distortion is still larger than that of the original data but other methods, such as $l$-diversity, can produce better results.

With regard to privacy protection, the proposed method resists the Spectral filtering (SPF) and Bayes-Estimated Data Reconstruction (BE-DR) attacks.
The proposed method was originally designed for small datasets. The next chapter introduces another data privacy perturbation algorithm that deals with a large volumes of data.
Chapter 6

μ-Fractal Data Perturbation

6.1 Introduction

This chapter presents a data privacy preserving technique named μ-Fractal data perturbation which is used for privacy protection of data publishing. It implements the data privacy preserving framework (DP²F, detailed in chapter 4), and incorporates fractals to take advantage of its self-similarity characteristic and chaotic feature (when the initial parameter is unknown) [181].
CHAPTER 6 $\mu$-FRactal DATA PERTURBATION

The evaluation and experiments are carried out in three categories, namely distribution and information content of the processed data. These methods have been introduced in chapter 5. Attack resistance tests are explained in the Appendix.

The main features of this chapter are the following. The proposed method (i) resists spectral filter (SPF) and Bayes-Estimation Data Reconstruction (BE-DR) attacks, ii) keeps the perturbed data in the same value range and data format as the original data, and iii) maintains the data distribution.

The rest of this chapter is organized as follows. Section 6.2 introduces the mathematical fundamentals of fractals and chaos. In section 6.3, the proposed method is detailed via perturbation values, the perturbation process and the restoration process. Section 6.4 describes experiments to evaluate the methods in terms of distribution and information added by processing the data. Section 6.5 discusses the proposed method against existing solutions and section 6.6 summarizes the chapter. Attach resistance tests are described in the Appendix.
CHAPTER 6 μ-FRACTAL DATA PERTURBATION

6.2 Mathematical Foundations

The adopted fractal function, called Bifurcation diagram of the logistic map, [182] is represented by equation (6-1), where \( \mu \) is a parameter in the fractal value sequence generation.

\[ x_{n+1} = \mu \cdot x_n (1-x_n) \]  \hspace{1cm} (6-1)

Figure 6.1 illustrates the fractal nature of the Bifurcation diagram of the logistic map [181], which shows the output sequences based on different \( \mu \) values. This is a typical fractal function and each sub-sequence of the fractal is similar to the overall sequence [181]. Also, in such a fractal, when \( \mu \) is between 3.5699 and 4, the system is chaotic [182, 183]. Chaotic here can be explained as “when the present determines the future, but the approximate present does not approximately determine the future” [181-183]. In other words, the chaotic fractal sequence heavily depends on the initial parameters and different initial parameters cannot derive the same fractal sequence.

To illustrate the fractal’s characteristics, Figure 6.2 (a-f) shows time sequences \( n \) in equation 6-1) based on different \( \mu \) values. It can be seen that the fluctuations in the fractal’s value become irregular when \( \mu \) is between 3.5699 and 4. This figure shows
the uncertainty of the fractal feature. NOTE: this chapter focuses on the fractal only when $\mu$ is greater than 3.5699 and less than 4.

*Figure 6.1: Bifurcation Diagram of The Logistic Map*

*Figure 6.2 (a): Time Sequences Based on $\mu=3.10000$*
Figure 6.2 (b): Time Sequences Based on \( \mu = 3.56990 \)

Figure 6.2 (c): Time Sequences Based on \( \mu = 3.80000 \)
CHAPTER 6 $\mu$-FRACTAL DATA PERTURBATION

Figure 6.2 (d): Time Sequences Based on $\mu=3.91230$

Figure 6.2 (e): Time Sequences Based on $\mu=4.00000$
The features used in this chapter are the fractal and chaos characteristics of this function. The fractal feature indicates self-similarity which means any part of the generated sequence has a similar shape to the overall sequence. The chaos feature indicates that without the initial values of the sequence, the whole sequence shows a random manner [181].

### 6.3 μ-Fractal Data Perturbation (μ-FDP)

This section first provides an overview of μ-FDP, including the overall process flow and algorithm brier. Then, it details the proposed algorithm in terms of the calculation.
CHAPTER 6 $\mu$-FRACTAL DATA PERTURBATION

of fractal sequences and perturbation vectors. At the end of this section, the restoration process is presented.

6.3.1 Overview of $\mu$-FDP

The main components of the perturbation process flow are the original data, fractal sequences 1 ($FS_1$) and 2 ($FS_2$), and perturbation vectors 1 ($PV_1$) and 2 ($PV_2$).

![Diagram of $\mu$-FDP Perturbation Process Flow](image1)

*Figure 6.3: $\mu$-FDP Perturbation Process Flow*

![Diagram of $\mu$-FDP Restoration Process Flow](image2)

*Figure 6.4: $\mu$-FDP Restoration Process Flow*
CHAPTER 6 $\mu$-FRACTAL DATA PERTURBATION

The main components in both perturbation and restoration flows are explained in the following.

1) *Original Data* denotes the attributes that are being protected, which can be a column in a micro-table or a vector. The data being protected have to be numeric, such as age, post code, disease code etc. As non-numeric data can be converted to numeric based on certain rules [169], this is not a real restriction. Typically, the type of original data has a determined format and value range, for example, an age should be within [1, 99], disease code should be within [1-001, 1-110] $\cap$ [2-001, 2-045] $\cap$ .... Usually, these two features, namely format and value range are used to justify whether the data is valid or not.

2) *Fractal Sequence (FS)* are the results of the fractal equations. Given the initial parameters and the total number of the results, a fractal sequence is derived. In the proposed method, there are two fractal sequences represented by $FS_1$ and $FS_2$. Both are generated from the equation (6-1) with different initial parameters. Each value in the sequences is within (0, 1) according to the feature of the applied fractal function [182].
3) Perturbation Vector ($PV$) is the noise sequence used to perturb the data. There are two perturbation vectors, $PV_1$ and $PV_2$ that are used as the multiplicative and additive portion of the perturbation noise, respectively.

To facilitate mathematical treatment, we assume that the data to be privacy-protected is in a row or a column in a micro-table. This data is used as a vector, and the perturbing calculations are executed on this vector. The perturbation, according to the $DP^2F$ (chapter 4), is performed in two steps: (i) noise calculation; and (ii) scaling. In the first step, two individual perturbation noise sets, $FS_1$ and $FS_2$, are calculated - one is for multiplicative perturbation and the other is for additive perturbation. The second step is scaling the $FS$ and generating the $PV$ to ensure that the perturbed data maintains data utility while protecting data privacy.

The perturbation can be written in the form of $PD = OD \times PV_1 + PV_2$, where $PD$ is the perturbed data, $OD$ is the original data, $PV_1$ and $PV_2$ are perturbation noise. $PV_1$ is a function of $FS_1$ and $PV_2$ is a function of $FS_2$.

To obtain the original data, the fractal sequences are recalculated, the scaling is reversed and the perturbation process is inverted. As both scaling and the proposed
perturbation are lossless operations, the original data can be accurately restored. The restoration process is detailed in section 6.3.3.

Table 6.1: Summarized Notion

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu_1$ and $\mu_2$</td>
<td>Fractal initial parameters for generating fractal sequence 1 and fractal sequence 2, see equation (6-1)</td>
</tr>
<tr>
<td>$x_0$ and $y_0$</td>
<td>Fractal initial parameters for generating fractal sequence 1 and fractal sequence 2, see equation (6-1)</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Scaling parameter for the multiplicative part of the proposed method</td>
</tr>
<tr>
<td>$\phi$</td>
<td>Scaling parameter for the additive part of the proposed method</td>
</tr>
<tr>
<td>$FS_1$ and $FS_2$</td>
<td>Fractal sequence 1 and fractal sequence 2, which are derived from equation (6-1)</td>
</tr>
<tr>
<td>$PV_1$ and $PV_2$</td>
<td>Perturbation vector 1 and perturbation vector 2, which are calculated from $FS_1$ and $FS_2$.</td>
</tr>
<tr>
<td>$A$ and $A'$</td>
<td>The original dataset and the perturbed dataset</td>
</tr>
<tr>
<td>$a_i$ and $a_i'$</td>
<td>The $i$-th data of the original dataset and of the perturbed data set</td>
</tr>
<tr>
<td>$M$</td>
<td>The number of data items in the original dataset</td>
</tr>
<tr>
<td>$p$ and $q$</td>
<td>The lower bound and upper bound of the original dataset</td>
</tr>
<tr>
<td>$g_i$ and $h_i$</td>
<td>The $i$-th element of perturbation vector 1 and perturbation vector 2</td>
</tr>
</tbody>
</table>

6.3.2 Perturbation Algorithm

The proposed algorithm has three parts, namely the generation of fractal sequences (subsection 6.3.2.1), the generation of the multiplicative perturbation vector (subsection 6.3.2.2) and the generation of the additive perturbation vector (subsection 6.3.2.3). Before introducing the algorithm, all symbols are summarized below.
6.3.2.1 Initial Parameters and Fractal Sequences

The parameters are used to initialize the perturbation process are the following: \( \mu_1 \) and \( X_0 \) for the generation of fractal sequence 1 (\( FS_1 \)), and \( \mu_2 \) and \( Y_0 \) for fractal sequence 2 (\( FS_2 \)). The algorithm operates progressively column-by-column. Here the explanation is given for one column (one attribute) and the extension of it to the other columns is trivial. The attribute can be represented as a vector \( A_i = [a_1, a_2, ..., a_M]^T \), where \( M \) denotes the total number of the original data, \( a_i \) denotes the \( i \)-th original data element, \( a_i \in [p, q] \) and \( i \in [1, M] \).

![Initial parameter](X_0)

![Initial parameter](\mu)

![Fractal Sequence](FS)

*Figure 6.5: FS Generation Flow*

In the proposed method, a value for \( \mu \) and for \( x_0 \) is chosen respectively, where \( \mu \in (3.5699, 4) \). With the parameters \( \mu_1 \) and \( x_0 \), the first fractal sequence can be calculated based on equation (6-1) and represented by \( FS_1 = [x_1, x_2, ..., x_M] \). Similarly, the second fractal sequence is calculated with the parameter \( \mu_2 \) and \( Y_0 \) and represented by \( FS_2 = [y_1, y_2, ..., y_M] \).
6.3.2.2 Perturbation Vector 1

The process of generating perturbation vector 1 \((PV_1)\) is depicted in Figure 6.6. The calculation is carried out in two steps, which are explained below.

\[ f : x_i \in (0, 1) \rightarrow x_i' \in [p, q] , \text{ where } i \text{ is from 1 to } M. \text{ The mapping equation is } x_i'(i) = x_i \cdot (q - p) + p. \]
The second step applies scaling parameter $\rho$ to all elements $g_i$ in $PV_1$ by
$$g_i = (a_i - x_i') \cdot \rho + a_i$$, where $a_i$ denotes the $i$-th original data element.

### 6.3.2.3 Perturbation Vector 2

The process of generating perturbation vector 2 ($PV_2$) is depicted in Figure 6.8. The calculation of element $h_i$ in $PV_2$ is
$$h_i = y_j \cdot \phi$$, where $\phi$ is the scaling parameter for additive part of the proposed method.
6.3.2.4 Perturbation

After the calculation of both perturbation vectors, the perturbed data is derived by combining these vectors (see Figure 6.9). Let \( a'_i \) be the perturbed data of \( a_i \). With both \( PV_1 \) and \( PV_2 \), the perturbed data \( a'(i) \) can be derived from \( a'_i = g_i + h_i \).
6.3.3 Restoration

The restoration key is the same to the perturbation key. In order to calculate the original data sets, the restoration key is composed of $x_0$, $\mu_1$ and $\rho$ for the calculation of $PV_1$, and $y_0$, $\mu_2$ and $\phi$ for the calculation of $PV_2$. Then the original data $a_i$ is calculated via the following steps:

- Both fractal sequence 1 ($FS_1$) and 2 ($FS_2$) are calculated based on equation (6-1)
- Perturbation vector 1 $PV_1$ is calculated as explained in section 6.3.2.2 and $PV_2$ is calculated as explained in section 6.3.2.3.
- Original data set $a_i$ is calculated based on equation (6-10)

$$a_i = \frac{a'_i - y'_i \cdot \phi + x'_i \cdot \rho}{\rho + 1} \quad \text{(6-10)}$$

6.4 Experiments and Results

This section starts with the evaluation methods used to examine the proposed method and then shows the experimental results.
6.4.1 Evaluation and Environmental Setup

6.4.1.1 Distribution

The number of original data samples was $M = 6400$, which was the upper limit of the test environment; the fractal sequence parameters were $x_0=0.1876$, $y_0=0.2859$, $\mu_1=3.8123$ and $\mu_2=3.7983$. For the proposed method, three experiments were conducted with parameters as follows:

i) $\rho=0.05$, $\phi = 3$, results in $PA \approx 6.8\%$.

ii) $\rho=0.1$, $\phi = 5$, results in $PA \approx 12.5\%$.

iii) $\rho=0.5$, $\phi = 12$, results in $PA \approx 22.4\%$.

These parameters resulted in perturbation amplitude $PA \approx 6.8\%$, 12.5% and 22.4% respectively. In the generalized method, the generalization interval was set to 5, such as [25, 30], [31, 35] and so on. The parameters for the proposed method were chosen as reasonable values for the generated dataset.

6.4.1.2 Information Content

The calculations of the added information content for the proposed method, $k$-anonymized data and $l$-diversity were introduced in the chapter 5. The test
environment was similar to that in chapter 5. Three sets of parameters were used to evaluate the distortion information content, which were the same as that in the distribution tests.

6.4.2 Experimental Performance

The experiments were conducted with regard to the quality of perturbation, i.e. to examine the distribution and the added distortion for the original data. Attack resistance results are explained in the Appendix.

6.4.2.1 Distribution Tests

Figure 6.10: Distribution Test -- Normal Distribution with PA ≈ 6.8%
Figure 6.11: Distribution Test -- Uniform Distribution with PA ≈ 6.8%

Figure 6.12: Distribution Test -- Normal Distribution with PA ≈ 12.5%
CHAPTER 6 \( \mu \)-FRACTAL DATA PERTURBATION

Figure 6.13: Distribution Test -- Uniform Distribution with PA \( \approx 12.5\% \)

Figure 6.14: Distribution Test -- Normal Distribution with PA \( \approx 22.4\% \)
CHAPTER 6 μ-FRACTAL DATA PERTURBATION

Figures 6.10 to 6.15 show the distribution of the original, perturbed and generalized data. As can be seen from the Figure 6.10 to 6.13, the distribution of the perturbated data is very close to the original data, while generalization changed the data distribution.

6.4.2.2 Empirical Information Content Tests

Figures 6.12 to 6.14 show the added information (distortion) from the proposed method, $k$-anonymity and $l$-diversity. As depicted, distortion is higher in the proposed method than in the other methods, even when perturbation amplitude ($PA$) is
relatively low. The figures also show that the added information from the proposed method increases as $PA$ increases.

Figure 6.12: Information Content Test with $PA \approx 6.8\%$
6.4.2.3 Attack Resistance Tests

Attach resistance tests are presented in the Appendix.

6.5 Discussion

One of the main features of the proposed method is that the \( \mu-FDP \) is able to effectively increase the added distortion to a very high level. The method also keeps the perturbed data within the same data format and value range as the original ones.
CHAPTER 6 $\mu$-FRACTAL DATA PERTURBATION

From the data distribution test, the proposed perturbation method maintains the distribution of the data, and the magnitude of the introduced change can be controlled, while generalization methods compromise on data utility and alter the distribution of the original data.

The data privacy attack experiments clearly show that the proposed method is able to resist both spectral filtering (SPF) and Bayes-Estimated Data Reconstruction (BE-DR) attacks which have successfully attacked many data transformation-based privacy protection techniques [145, 162]. The proposed method also ensures that no matter what the original data value(s) is, the perturbed data are able to keep the perturbed values in the same value range as the original. The significance of this feature is that the perturbed data cannot be distinguished from the original data. The perturbation parameters control the perturbing effects in the process.

The proposed method requires more parameters compared with the approach proposed in chapter 5, but provide a better support for large volume of data.
6.6 Summary

This chapter presents an effective method for data perturbation to provide privacy protection of numeric data. The effectiveness of the data perturbation method lies in the fact that it is based on fractal and chaos theory to derive perturbation vectors. A distinguishing contribution of the proposed method is that it provides maximal utility for public data analysts who do not have a restoration key while at the same time, it protects sensitive data from data linkage attacks (discussed in chapter 4). The usefulness of our algorithm was shown by conducting detailed experiments to demonstrate its impact on both the data perturbation and maximal data utility features. The results of the experiments also showed that our perturbation algorithm could be applied as desired on data with different distributions, namely uniform and normal.
Chapter 7

Conclusion and Future Work

This thesis looked at two privacy preserving data sharing approaches: data sharing with known recipients (chapters 2 and 3) and with unknown recipients (chapters 4 to 6).

For sharing data with known recipients, a privacy preserving access control model was proposed. The model addressed three challenges, namely i) the capability to cater for unique users, ii) multiple privileges, conditional and sequential, controlled privileges, and iii) user roaming and data roaming in cross-domain environments. The proposed functional module in chapter 2 contains a system framework based on a label-based mechanism. The hybrid role-based and attribute-based user privilege control supports unique users, and a hierarchical control structure on both user and
CHAPTER 7 CONCLUSION AND FUTURE WORK

data servers is proposed for diverse privileges, conditional privileges and operation sequences. The designation of a collaborating subject server and an object server gives more crafted and flexible capability for subject roaming, object roaming and identifying responsibility of data management in roaming scenarios. A label-based privilege refinement mechanism enables the management of privilege hierarchies.

Next, the thesis investigated the problem of involving access purpose in the privacy preserving model and addressed two challenges: the involvement of purpose in both user control and granular data control, and purpose translation in cross-domain environments. Chapter 3 presented a hierarchical subject purpose control method to handle user purpose, user obligations and server constraints for unique users. It has an object classification and description mechanism to deal with object related purpose that is made up of data owner’s intended purpose, conditional purpose and object obligations. A purpose translation layer on both subject server and object server provides consistent translation of the same purpose between different domains. With incorporating the two functional modules proposed in chapters 2 and 3, the overall access control model is able to deal with complex, collaborating and organizational environments, and can be instantiated for practical applications.
When sharing data with unknown recipients, which is also called data publishing, the challenge was to keep data privacy while optimizing data utility. Data privacy is preserved if authorized receivers can obtain the original data, while others can only access processed data and an adversary cannot derive the original data, or the reconstructed results are not close enough to the original data. Utility can indicate data distribution, data format, or data range.

To balance data privacy and data utility, the thesis proposed two hybrid data privacy algorithms that combine additive and multiplicative perturbation. The algorithm in chapter 5 was built on Chebyshev polynomials, and it generated two sequences, one for the additive and one for the multiplicative step. The involvement of scaling parameters ensured the perturbed values were in the same data range as the original data.

The effectiveness of perturbation was examined by looking at entropy, comparison of distributions and performing two, previously published, classic data privacy attacks. The attack methods introduced in chapter 4 were Spectral Filter (SPF) and Bayes-Estimated Data Reconstruction (BE-DR). The entropy results showed that the proposed algorithm significantly increased the entropy of the data and outperformed $k$-anonymity and $l$-diversity when the magnitude of the perturbation amplitude was at
least 40% of the original data. Even when the noise was only 10%, the proposed method still outperformed $k$-anonymity. The distribution tests showed that for two common distributions, normal and uniform, the method proposed in chapter 5 was able to maintain data utility. The RMSE results showed that the two classic data reconstruction attacks were not able to reconstruct the original data from the perturbed data.

The algorithm in chapter 6 was built on a fractal, called Bifurcation diagram. Again, an additional and multiplicative hybrid scheme was used to incorporate the two generated fractal sequences. Scaling parameters were used to control the perturbation and keep the perturbed data similar to the original data.

The method proposed in chapter 6 was examined the same way as in chapter 5. The entropy test results showed that the proposed algorithm increased the entropy of the data. The results outperformed $k$-anonymity and $l$-diversity for the magnitude of the perturbation amplitude was at least 5% of the original data. The distribution tests showed that when the original data was following either of two common distributions, normal and uniform, the distribution of the perturbed data was close to the original’s. The RMSE results showed that the two classic data reconstruction attacks were not successful.
CHAPTER 7 CONCLUSION AND FUTURE WORK

Future work can look at devising a high level language for the proposed model in Part I so that it would be easier to deploy to existing organizations; building proper user management based on the proposed model for contemporary authentication server databases and evaluating the instantiation performance of privilege refinement in terms of runtime complexity and memory consumption.

Future work to Part II may investigate the application of the proposed method to multiple data attributes at one time; applying the proposed methods to information hiding; evaluation of the performance in terms of runtime complexity for different scenarios.
Bibliography


[12] Gross, Ralph, and Alessandro Acquisti. "Information revelation and privacy in


large data bases. VLDB Endowment, 2006.


[38] RBAC Ocial Website. An Introduction to RBAC. Available at http://csrc.nist.gov/rbac/


[81] LeFevre, Kristen, David J. DeWitt, and Raghu Ramakrishnan. "Mondrian


[121] Farzad, Faranak, Eric Yu, and Patrick CK Hung. "Role-Based Access Control


[148] Hong, Dowon, and Abedelaziz Mohaisen. "Augmented Rotation-Based


Liu K., Multiplicative data perturbation for privacy preserving data mining, Dept. of Computer Science and Electrical Engineering University of Maryland, Baltimore County (UMBC), 2007


Appendix A

Attack Resistance

1. Description of the Attacks

This section introduces two classic attack methods that are used to evaluate the proposed data perturbation methods in terms of attack success. The two attacks were evaluated and summarized in [162].

The assumptions in both attacks are as follows [145].

i) The introduced noise is random, has a zero mean ($\mu_{\text{noise}}=0$).

ii) The original data, noise, and perturbed data are in the form of an $R$ by $C$ matrix respectively, and $R = C$ to facilitate experiments and comparison of different attack results [162].

iii) The original data set $O$ is square and the elements in the original dataset ($O$) are independent of each other, so that $O$ and the covariance of the original data ($O_{\text{cov}}$) have distinct and non-zero eigenvalues. This assumption holds in most practical situations [162].
iv) The noise dataset matrix and the original dataset are uncorrelated and the distributions of the noise dataset matrix and the perturbed dataset matrix are known to the attacker [162]. Since the attacker does not know the distribution of the original data, the initial assumption is that the original dataset follows a normal distribution.

v) The perturbed data ($\tilde{P}$) is public.

1.1 Spectral Filtering

The first attack used for resistance testing is called spectral filtering (SPF) [145, 173], and is a random matrix-based approach for reconstructing the original dataset from the perturbed data. Arranging the perturbed data in a matrix, the eigenvalues of this matrix are used to estimate the introduced noise, while assuming the noise has a normal distribution. The attack result has two parts: i) the estimated original dataset matrix and ii) the estimated noise matrix [135, 162].

The calculation steps were presented in [145], evaluated in [162] and are briefly summarized as follows. According to the assumption, the noise distribution is known and the variance is $\sigma^2$. First the eigenvalues of the covariance matrix of the perturbation noise $\lambda_{\text{min}}$ and $\lambda_{\text{max}}$ are calculated based on $\lambda_{\text{min}} = \sigma^2 (1 - 1/\sqrt{Q})^2$ and $\lambda_{\text{max}} = \sigma^2 (1 + 1/\sqrt{Q})^2$, where $Q$ represents the asymptotic value of $M/N$ when the
number of data samples approaches infinity[145]. In the experiments, \( Q=1 \) [162]. The the eigenvalues \( \lambda_i \) of the covariance matrix \( \tilde{P}_{cov} \) are computed, and the noise eigenvalues satisfying \( \lambda_i \geq \lambda_{\text{min}} \) and \( \lambda_j \leq \lambda_{\text{max}} \) are identified [155].

The metric used to evaluate the success of an SPF attack is the Root Mean Square Error (RMSE) that measures the difference between the original \((o_i)\) and reconstructed \((\tilde{o}_i)\) data, i.e. \( \text{RMSE} = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (o_i - \tilde{o}_i)^2} \). An RMSE value between 0 and 1 indicates that attackers have a high probability of reconstructing the original data \( O \). If \( \text{RMSE} = 0 \) then \( O \) has been accurately reconstructed; if \( \text{RMSE} \) is equal or greater than 1, then it means no data in \( O \) has been recovered.

### 1.2 Bayes-Estimated Data Reconstruction

The second data reconstruction method is called Bayes-Estimated Data Reconstruction (BE-DR) which is proposed in [92] and evaluated in [162].

The BE-DR attack steps are briefly summarized as follows. Calculate \( O_{cov} \) by \( O_{cov} = \tilde{P}_{cov} - \sigma^2 \) for all elements in \( \tilde{P}_{cov} \) and derive the mean vector \( \mu_O \) of the original data matrix from the mean vector \( \mu_{\tilde{P}} \) of perturbed data matrix by \( \mu_O = \mu_{\tilde{P}} \) (according the assumption i) that the noise mean is zero). Then, the estimated original
APPENDIX A
ATTACK RESISTANCE

data elements are calculated by $\tilde{\alpha}_y = (O_{\text{cov}}^{-1} + 1/\sigma^2 I)(O_{\text{cov}}^{-1} \mu_o + \tilde{P} / \sigma^2)^{-1}$ [162]. The success of the EB-DR attack is measured by the RMSE explained in SPF attack.

2. Attack Results

A 6400-item dataset was generated to test the attack resistance of the proposed method. The data was stored in vector form and then re-formed to matrix form to be easier to implement attack methods.

In each data reconstruction case, the attack method obtained an estimated data set which was then evaluated by measuring the Root Mean Square Error (RMSE).

2.1 Chebyshev Data Perturbation

For the attack resistance test, two classic data reconstruction attacks were used, namely spectral filtering (SPF) and Bayes-Estimated Data Reconstruction (BE-DR).

Spectral Filtering

This test shows the result of the special filtering (SPF) for varying numbers of data samples. For each data set, the Root Mean Square Error (RMSE) test was executed five times and the average was calculated. In this test, every RMSE was far greater
APPENDIX A
ATTACK RESISTANCE

than 1 and the average was 26.1621, so the attack was considered not able to attack the proposed method. Note: RMSE varied between test runs.

Figure 1: RMSE for SPF Attack

Bayes-Estimated Data Reconstruction

This test was executed on the same number of data samples, used RMSE to evaluate the success of the attack. The RMSE result of BE-DR was also far greater than 1, the average was 23.3614. Therefore, the attack was not successful.
To summarize the evaluations, the proposed method is able to maintain the data distribution, brings higher added entropy than the compared method and also resists to SPF and BE-DR attacks.

### 2.2 μ-Fractal Data Perturbation

Figure 3 shows the SPF attack results. The mean square error (MSE) is more than 39, which is far above the acceptable value 1, and this means the attack failed to re-construct the original data. Figure 4 shows the BE-DR attack result. The root mean square error (RMSE) is more than 23 which is again far above the acceptable value of 1, which means the attack failed to re-construct the original data.
APPENDIX A
ATTACK RESISTANCE

Spectral Filter (SPF)

Figure 3: SPF Attack Result

Bayes-Estimated Data Reconstruction (BE-DR)

Figure 4: BE-DR Attack Result
Appendix B

Program Code--Experiments of Data Perturbation Methods

%Created by Jian Zhong

%Contact email: jian.zhong@rmit.edu.au

%You are free to use this code but you cannot remove the author's info.

function cdpDiff(M,k,n)

%orgdata=zeros(M);

alpha=4.7; %shift

beta=(-1+n+6)/(n+6);

groupindex=zeros(k);

gamma=1.5;

meancdp=0;

RMS=0;

%M=100;
APPENDIX B
PROGRAM CODE

MEAN=39;
SD=3;
ptSD=0;
ptvar=zeros(M);
%orgdata=zeros(M);
%orgdataX=normrnd(MEAN,SD,M,1);
%orgdataX = random('norm',MEAN,SD,M, 1);
orgdataX = randi([25,65],1,M);

ptdata =zeros(M,1);
cdpt=zeros(M,1);
cdpt2=zeros(M,1);
pt=zeros(M,1);
diff=zeros(M,1);

orgfid = fopen('orgdata.txt','w');
fprintf(orgfid,'%f
',orgdataX);
fclose(orgfid);

fid=fopen('orgdata.txt','r');
APPENDIX B
PROGRAM CODE

for i=1:M

    orgdata(i)=fscanf(fid,'%f
',1);

end

% A=fscanf(fid,'%f')

% size(A,1)
fclose(fid);

DL=M/k;
t=zeros(k);
t0=0;
t(1)=floor(DL);
groupindex(1)=0;
for i=2:1:k
    t(i)=floor(i*DL);
    for c=1:1:i-1
        t(i)=t(i)-t(c);
    end
end
for a=2:1:k
for j=1:1:a-1
    groupindex(a)=t(j)+groupindex(a);
end
end

T1=ChebT(n);
T2=ChebT(n+1);

parafid = fopen('para.txt','w');
fprintf(parafid,'M = %i ; 
',M);
fprintf(parafid,'k = %f ; 
',k);
fprintf(parafid,'n = %i ; 
',n);
fprintf(parafid,'Alpha = %f 
',alpha);
fprintf(parafid,'Beta = %f 
',beta);
fprintf(parafid,'Gamma = %f 
',gamma);
fprintf(parafid,'DL = %f 
',DL);
fprintf(parafid,'t = %i ; 
',t);
fprintf(parafid,'GroupIndex = %i ; 
',groupindex);
fprintf(parafid,'Tn = %i 
',T1);
fclose(parafid);
APPENDIX B
PROGRAM CODE

%disp(cdpt);

for b=1:1:k
    %groupindex(i)=t(i)+groupindex(i);
temp=0;
    for j=1:1:t(b)
        x=-beta+(2*beta*j)/(t(b)+alpha);  %CHANGED to cal cdpt it
        is ok to use -beta, I also change the beta vaule accordingly.
            for p=1:1:n

        %cdpt(groupindex(b)+j)=cdpt(groupindex(b)+j)*(x^p)*T(n-p+1)+(x ^p)*T((n+1)-p+1);

        cdpt(groupindex(b)+j,1)=cdpt(groupindex(b)+j,1)+(x^p)*T1(n-p+1) -floor(cdpt(groupindex(b)+j,1)+(x^p)*T1(n-p+1));
        %temp=temp+cdpt(groupindex(b)+j);
    end
for q=1:1:n+1

cdpt2(groupindex(b)+j,1)=cdpt2(groupindex(b)+j,1)+(x^q)*T2((n+1)-q+1)-floor(cdpt2(groupindex(b)+j,1)+(x^q)*T2((n+1)-q+1));

end

%temp=temp+cdpt(groupindex(b)+j);

%meancdp=floor(temp*1000)/t(b);

%if cdpt(groupindex(b)+j)>=0 && cdpt2(groupindex(b)+j)>=0

%pt(groupindex(b)+j,1)=mod(floor(gamma*cdpt(groupindex(b)+j,1)*100),11);

%pt(groupindex(b)+j,1)=floor(gamma*cdpt(groupindex(b)+j,1));

%else

%pt(groupindex(b)+j,1)=-mod(floor(gamma*cdpt(groupindex(b)+j,1)*100),11); %same mod as above

%pt(groupindex(b)+j,1)=-mod(floor(gamma*cdpt(groupindex(b)+j,1)

));
%pt(groupindex(b)+j)=mod(floor(cdpt(groupindex(b)+j)*100),SD);
%temp=temp+pt(groupindex(b)+j,1);
%meancdp=temp/t(b);

%ptdata(groupindex(b)+j,1)=orgdata(groupindex(b)+j)*(cdpt(groupindex(b)+j,1)*0.05+0.95)-floor(orgdata(groupindex(b)+j)*(cdpt(groupindex(b)+j,1)*0.05+0.95)+cdpt2(groupindex(b)+j,1)+floor(pt(groupindex(b)+j,1)-meancdp);

ptdata(groupindex(b)+j,1)=orgdata(groupindex(b)+j)*(cdpt(groupindex(b)+j,1)*0.01+0.99)+cdpt2(groupindex(b)+j,1)*gamma;

pt(groupindex(b)+j,1)=ptdata(groupindex(b)+j,1)-orgdata(groupindex(b)+j);

RMS=RMS+pt(groupindex(b)+j,1)*pt(groupindex(b)+j,1);

diff(groupindex(b)+j,1)=abs(pt(groupindex(b)+j,1))/orgdata(groupindex(b)+j);
APPENDIX B
PROGRAM CODE

    end

    end

RMS=sqrt(RMS/M);
disp(RMS);
%disp(cdpt);
%disp(cdpt2);
%disp(pt);
%disp(T1);
%disp(T2);
%disp(diff);
ptvar=var(pt(:,1));
%disp(ptvar);
%for o=1:1:M
%    ptSD=ptSD+ptvar(o);
%end

%for o=1:1:M
APPENDIX B
PROGRAM CODE

ptSD=ptSD+ptvar;
%end

%disp(ptSD);

ptSD=ptSD/M;

midfid = fopen('mid.txt','w');
fprintf(midfid,'ptSD = %f \n',ptSD);
fprintf(midfid,'cdpi(j) = %f \n',cdpt);
fprintf(midfid,'meancdp = %f \n',meancdp);
fprintf(midfid,'pti(j) = %i \n', pt);
fprintf(midfid,'ptdata = %i \n',ptdata);
fclose(midfid);

compfid = fopen('comp.txt','w');
for d=1:1:M
    fprintf(compfid,'%f \n',orgdata(d));
    fprintf(compfid,'%f \n',ptdata(d,1));
end
fclose(compfid);
APPENDIX B
PROGRAM CODE

```matlab
pertrbfid = fopen('pertrb.txt','w');

for d=1:1:M
    fprintf(pertrbfid,'%f
' ,ptdata(d,1));
end

close(pertrbfid);

%ptmean=mean(ptdata);

yorgdatamin=min(diff);
yorgdatamax=max(diff);

yptdatamin=min(diff);
yptdatamax=max(diff);

xorgdata=linspace(yorgdatamin,yorgdatamax,10);

yyorgdata=hist(orgdata,xorgdata);

yyptdata=hist(diff(:,1),xorgdata);

yyorgdata=yyorgdata/M;

yyptdata=yyptdata/M;
```
hold on

plot(xorgdata,yyptdata,'c','LineWidth',3);

hold off